

ADA 033102

12  
B.S.

R-1872-PR  
September 1976

# AIDA: An Airbase Damage Assessment Model

D. E. Emerson

DDC  
RECEIVED  
DEC 9 1976  
A

A report prepared for  
UNITED STATES AIR FORCE PROJECT RAND

DISTRIBUTION STATEMENT A  
Approved for public release;  
Distribution Unlimited

**Rand**  
SANTA MONICA, CA. 90406

COPY AVAILABLE TO DDC DOES NOT  
PERMIT FULLY LEGIBLE PRODUCTION

Copy available to DDC does not  
permit fully legible reproduction

The research described in this report was sponsored by the United States Air Force under Contract No. F44620-73-C-0011 — Monitored by the Directorate of Planning, Programming and Analysis, Deputy Chief of Staff, Research and Development, Hq USAF.

Reports of The Rand Corporation do not necessarily reflect the opinions or policies of the sponsors of Rand research.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

14

1. REPORT NUMBER  
R-1872-PR

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

6

4. TITLE (and Subtitle)  
AIDA: AN AIRBASE DAMAGE ASSESSMENT MODEL

5. TYPE OF REPORT & PERIOD COVERED  
9 Interim Report

6. PERFORMING ORG. REPORT NUMBER

10

7. AUTHOR(s)  
D. E. Emerson

15

8. CONTRACT OR GRANT NUMBER (if any)

F44620-73-C-0011

9. PERFORMING ORGANIZATION NAME AND ADDRESS

The Rand Corporation  
1700 Main Street  
Santa Monica, Ca. 90406

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

FCRC Office (AF/RDXTR)  
Director of Planning, Programming & Analysis  
Hq USAF, Washington, D.C. 20330

11

12. REPORT DATE

September 1976

13. NUMBER OF PAGES

88

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

12 95p.

15. SECURITY CLASS. (of this report)

UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

No restrictions

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Damage Assessment  
Military Air Facilities  
Bombardment (Attack)  
Targets

Computer Programs  
AIDA

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

see reverse side

7/B

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

↓  
Describes the airbase damage assessment (AIDA) computer model, designed for rapid examination of the results of conventional air attacks on complex targets. The model incorporates both a Monte Carlo and a deterministic mode of operation and has a variety of possible applications. It can be used to help plan effective attacks against complex target sets, or--by testing alternative attack headings, aim points, and the like--to examine the tradeoffs between damage to primary and secondary targets. It should also prove useful in airbase protection studies, both to provide realistic and detailed samples of possible damage patterns and to assess different options. Damage statistics for up to 250 individual targets can be quickly assessed for attacks involving as many as 50 delivery passes and 10 types of weapons. Both point-impact and area weapons can be handled, and targets may be grouped into 20 different vulnerability categories to distinguish different levels of weapon effectiveness. A complete user's guide and program listing are included. (PB)

UNCLASSIFIED



R-1872-PR  
September 1976

# AIDA: An Airbase Damage Assessment Model

D. E. Emerson

A report prepared for  
UNITED STATES AIR FORCE PROJECT RAND

**Rand**  
SANTA MONICA, CA. 90406

PREFACE

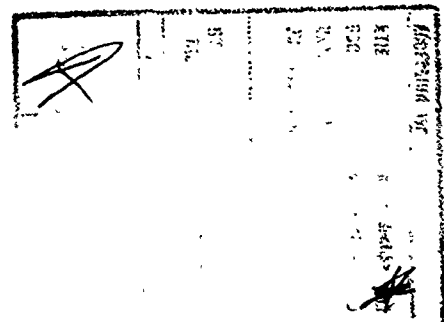
This report describes the Airbase Damage Assessment (AIDA) computer model, designed for rapid examination of the results of conventional air attacks on complex targets. A user's guide and program listing are included.

AIDA was developed by Rand for use in a study, conducted at Hq USAFE, of conventional air attack effectiveness. The model incorporates both a Monte Carlo mode and a deterministic (or expected-value) mode of operation. Approximations used in the expected-value computations, while differing from those used by the Joint Munitions Effectiveness Manual (JMEM), are equivalent in precision to those of the JMEM hand methods. A large body of publications furnishes background for the methods used and for input values required by AIDA.

AIDA has a variety of possible applications. It can be used as an aid in planning effective attacks against complex target sets, or--by testing alternative attack headings, aim points, and the like--in examining the tradeoffs between damage to primary and secondary targets. It should also prove useful in target (e.g., airbase) protection studies, both to provide realistic and detailed samples of possible damage patterns and to assess different protection options.

AIDA has been discussed with, and made available to, the Operations, Intelligence, Plans, Logistics, Engineering, and Communications staffs at Hq USAFE; to the staff of the Assistant Chief of Staff for Studies and Analysis at Hq USAF; and to a number of other DoD and NATO organizations. This report is being published to provide a record of the model and to make it available to a wider audience. The computer program is available from The Rand Corporation.

This work was conducted under the Project RAND research project entitled "Rand Ramstein Activity."



SUMMARY

This report describes a new computer model that permits examination of conventional bombing attacks on complex targets--e.g., on an airfield. A complete user's guide is included. Damage statistics for up to 250 individual targets can be quickly assessed for attacks involving as many as 50 delivery passes and 10 types of weapons. Both point-impact weapons (such as general-purpose (GP) bombs and precision-guided munitions (PGMs)) and area weapons--cluster bomb units (CBUs)--can be handled, and targets may be grouped into 20 different vulnerability categories to distinguish different levels of weapon effectiveness. If the user is concerned only with the expected numbers of hits with point-impact weapons, and is not interested in either CBU weapons or the coverage and damage variations expected with point-impact weapons, a special, more efficient expected-value mode is provided.

In its basic mode, AIDA determines the actual impact points (pattern centroids for CBUs) by Monte Carlo procedures--i.e., by random selections from the appropriate error distributions. GP bombs and PGMs that impact within a specified distance of a target are classed as hits, and the results include the total number of hits on each target and the cumulative probability of kill. For CBU munitions the program assesses the fraction of each target covered by each pattern, and the results include the fractional coverage from all patterns and a cumulative probability of kill for each target. In addition to these results for the complete attack, the attack can be repeated automatically for several trials to provide statistics on the average damage levels to be expected. In the special expected-value mode, average hit densities are determined directly, without recourse to Monte Carlo procedures.

In the basic mode, up to 5 targets may be designated as runways or taxiways suitable for aircraft operations, and the model will examine these to see if an area of specified size is available for such operations; if not, the minimum number of craters that would need to be repaired to obtain an area of that size is determined.

The AIDA program is designed so that many cases can be examined conveniently during each run--a feature that will prove useful, for

example, when one desires to quickly examine several attack options. Computer output includes both the input data and the attack results that have been specified. Most model features are illustrated with a sample problem.

CONTENTS

PREFACE .....	iii
SUMMARY .....	v
Section	
I. MODEL DESCRIPTION .....	1
Monte Carlo Mode .....	2
Expected-Value Mode .....	5
AIDA Operation .....	6
II. INPUT PROCEDURES .....	8
III. PROGRAM NOTES .....	15
IV. SAMPLE PROBLEM .....	18
Input .....	18
Output .....	23
Appendix	
A. DETAILED DESCRIPTION OF AIDA INPUT .....	37
B. GLOSSARY OF TERMS USED IN AIDA .....	51
C. PROGRAM LISTING .....	57

## I. MODEL DESCRIPTION

The AIDA (Airbase Damage Assessment) computer model permits examination of bombing attacks on a complex set of targets--e.g., on an airfield. In AIDA's basic mode of operation, the actual bomb impact points are obtained by Monte Carlo procedures,\* and the attack can be repeated for several trials to provide statistical estimates of the average damage and variability of that damage for each of the many targets. Alternatively, an expected-value mode is offered when only point-impact weapons are employed in the attack and when the user is interested only in the expected numbers of hits. With a variant of the expected-value mode, one may also generate hit-density patterns for complex attacks without specifying an actual target system. Both modes of operation and several different sets of problems may be treated by successive cases during a single computer run. The several features available with AIDA are illustrated in Sec. IV.

In AIDA the target system may be composed of up to 250<sup>†</sup> separate targets, for example, shelters, hangers, maintenance buildings, runways, taxiways--even pipelines. The complete attack may consist of up to 50 distinct weapon-delivery passes. Each target is a rectangle of specific size and orientation and an attack pass is defined by the expected probability of arrival, a heading, and the aim point, delivery accuracy, and dispersion for a stick of weapons. Targets may be grouped into 10 or 20 different vulnerability categories and there may be up to 10 different kinds of weapons dropped in an attack.

This computer model has a variety of possible applications. Since an entire airbase may be represented rather accurately, AIDA can be used

---

\* That is, the actual mean point of impact and the actual impact points are determined by random variates drawn from the appropriate error distributions.

<sup>†</sup> Most array dimensions may be changed fairly readily, and they have been in several instances. Some of the dimensions mentioned in this report differ from those outlined in an earlier version of AIDA; the changes were made to maximize program utility while limiting core storage requirements. Users will find instructions for changing program dimensions at the beginning of AIDA's MAIN subroutine.

to study how one might best design an attack to damage both primary and secondary targets. Whereas many bomb-damage aids deal only with a single target (or set of like targets) and provide no direct evidence as to the collateral damage to be expected, AIDA could be used to help design an attack that would get the maximum benefit from those bombs that miss the primary target. AIDA should also prove useful in a variety of studies of base protection and repair, in that it could provide a more realistic and detailed picture of possible damage than is customarily available. Yet another area in which AIDA may prove useful is as the first step in an analysis of the sortie-generation capabilities of a damaged base. Hopefully the results of this model can be used as input to a sortie-generation model,\* and the combination may then be able to provide a much better understanding of the effects of attacks on aircraft operations than exists today. In addition to analyzing problems involving airfield complexes, AIDA could just as easily be used to assess attack options and expected damage levels for attacks against the complex target arrays found at field headquarters, SAM sites, supply depots, etc.

Although AIDA is straightforward and simple mathematically,† its substantial flexibility should prove useful in a wide range of studies. For the reader who wants more detail on the methodological treatment than is discussed in the body of this report, a program listing is included in Appendix C.

#### MONTE CARLO MODE

In this mode of AIDA, weapons may be of two basic types: point-impact weapons (such as general-purpose (GP) bombs and precision-guided munitions (PGMs) or area weapons (such as cluster bomb units (CBUs)). A weapon reliability may be associated with each kind of weapon. For each kind of point-impact weapon an effective miss distance (EMD) may be specified for each target type (i.e., that miss distance at which the weapon is effective and an impact is to be categorized as a hit).

---

\*One that can account for the effects of various levels of damage to the various elements of a base.

†Approximations used in the AIDA computations, while different from those used in the Joint Munitions Effectiveness Manual (JMEM), are about equivalent in precision to those of the JMEM hand methods.

When this is done, target coverage is computed as that fraction of the target area that is covered by a circle having a radius of EMD and centered at the impact point. Additionally, the user may specify a value for target kill probability, given a hit (as defined above), or he may specify a different radius than the EMD and that will also be used for computing a value of coverage. This option may be specified individually for each target type and weapon type, as with the EMD. For example, with this feature one can use the EMD to represent the radius of the mean area of effectiveness (MAE) for severe structural damage to a building, and the alternate value to represent the radius of the MAE for severe damage to the building's contents, given a hit on the structure. The final results will include the cumulative coverage fractions for each target for all point-impact weapons, corresponding to both the EMD and to the optional factor, computed according to the relation:

$$FC = 1 - \prod_{\text{all hits}} (1 - C_{w,t}(i)) ,$$

where in the first case,  $C_{w,t}(i)$  is the target coverage based on the EMD of the  $i$ th hit on the target of type  $t$  by a weapon of type  $w$ ; in the second, it is assessed according to the user specified option.

If weapons are specified as CBUs, the model first computes the fraction of the area of each target that is covered by the rectangular bomblet pattern. The total fractional coverage of a target for all passes is that fraction of the area that has been covered by one or more patterns. If a probability of kill is associated with "coverage" by CBU patterns, the model will generate the total probability of kill, taking into account the actual position of each of the CBU patterns that covered any portion of the target, according to the relation:

$$PK = 1 - \sum_{i=1}^T \left\{ \prod_{a=1}^M (1 - p_{k,w,t})^{N(a,i)} \right\} / T ,$$



where  $a$  = the attack number,  
 $i$  = a point on the target grid,  
 $N(a,i)$  = the number of times point  $i$  was "covered" during  
attack  $a$ ,  
 $M$  = the total number of CBU attacks,  
 $T$  = the total number of target grid positions, and  
 $p_{k,w,t}$  = the probability of kill of a portion of a target  
of type  $t$  that is "covered" by a bomblet pattern  
from a weapon of type  $w$ .

The results for each trial include the number of hits by point-impact weapons and the fractional coverage by CBUs for each target as well as the point-impact weapon coverage (FC) and CBU kill probability (PK); in addition, for the targets that the user has specified (for a maximum of 20 targets other than the runways and taxiways), the impact points and weapon types are printed for up to 25 weapons per target. For multiple trials, the results for each target include the fraction of trials with at least one hit, the average number of hits and average CBU coverage, the standard deviation of these two measures, and the average values of FC and PK for the several trials. A full description of the output options is presented in the next section and is summarized in Table 1, p. 9.

The user may also specify\* that certain (up to 5) of the (rectangular) targets are actually runways or taxiways that are suitable for aircraft operations. The model will then test to see if such operations are possible from these areas; i.e., tests are made to see if the minimum clear length and width required for operations are available after the attack. In testing for runway availability, only point-impact weapons are considered, and the crater radius is taken as the EMD. Up to 250 hits can be stored and examined for each such target (the locations will be listed along with hit data on other targets). If the runway does not meet the minimum requirements, the user may request an assessment of how few craters would need to be repaired to meet those requirements. The user may also request an approximate computer plot of the impact points for each runway.

---

\*By defining as type #1 targets.

Coverage by CBU patterns is not taken into account in the runway assessments;\* examination of the effects of CBUs on the runway would require that the user either make a visual estimate using the approximate computer plot or plot the impact points and the resulting bomblet patterns more accurately on a plan of the airbase.

#### EXPECTED-VALUE MODE

This mode operates with target and attack descriptions that are identical to those used with the Monte Carlo mode of operation. However, only point-impact weapons may be included in the attack. Computationally this mode derives an average value of the hit density<sup>†</sup> for each target and for each attack. These are combined to provide the total expected number of hits for all the attacks. Although this mode provides no evidence regarding the statistical variations that must be expected in actual hit patterns, it does provide a quick, efficient means of assessing the expected values. Since, for many problems, a rather large number of Monte Carlo trials is required in order to get a reasonably accurate estimate of average values, one can sometimes use both the Monte Carlo and expected-value modes to advantage. The Monte Carlo mode can first be used with a limited number of trials to provide gross estimates of target coverage and of the variability of hits; the expected-value mode can then speedily provide a reasonably accurate estimate of the average number of hits. As will be illustrated

---

\* Except that the centroids are shown on the computer plots when they fall within the area included in the plot.

† The point value of the hit density is first determined at several points on each target for the specified aiming errors and dispersion; these values are computed for each of several positions along the intended line of bomb impacts. The average hit density is taken as the arithmetic average of these point values for the several target points and the several positions along the bomb impact line. If the target dimensions are small (i.e., less than one-quarter the RMS of the range and deflection errors projected parallel to each target edge), only the target corners are used for averaging. For larger targets, a uniform grid of internal points is established with spacing no greater than one-quarter the RMS noted above. The expected number of hits is computed as the product of the average hit density and the area of the target (including a border as wide as the EMD).

in Sec. IV, this two-step calculation would require only two more cards than would either calculation alone.

A special feature of the expected-value mode permits the user to quickly generate a hit-density grid. This feature is controlled by special target cards that may be used either alone, or in conjunction with normal target cards. Each such card generates a  $17 \times 17$  grid of hit-density values measured over a square of specified dimensions (the dimension should be a multiple of 16). If no dimension is specified, hit densities are provided at 250 ft intervals over a  $4000 \times 4000$  foot square. The southwest corner for each grid is placed at that position specified on the special target card. Such cards are identified by specifying target type #21--an entry that acts as a control signal within AIDA.

#### AIDA OPERATION

AIDA offers several features designed to simplify its operation and to permit a series of cases to be analyzed during a single computer run. Most are illustrated in Sec. IV. The first feature permits a multi-aircraft attack against the same objective to be specified simply; when two or more attacks have common parameters (i.e., heading, desired mean point of impact (DMPI), CEP, dispersion, arrival probability), a single entry will generate the additional attacks. Other convenience features derive from use of the REDO card (see the next section). When this card is encountered it acts as a terminator card, terminating the input for one case and announcing that there will be a subsequent case. If desired, an entirely new set of data may be input following a REDO card. Alternatively, the subsequent case may be a modification of the preceding case; a few simple inputs permit the user a wide range of modification options. One may change AIDA's mode of operation or any of the control variables. In addition, targets may be added to the prior list; or some may be removed and then others added. The same can be done with attacks. For example, this feature could prove particularly useful in investigating a series of alternative attack options on a given airfield.

AIDA's features provide substantial flexibility. For example, since target location is not restricted, it is possible to have two identical targets at the same location; by assigning these targets different type numbers, one can assess results for two weapon effects. This would be useful, for example, in determining expected personnel and materiel losses (targets with different MAEs) in open parking areas. The kill probability options available with impact weapons complement this technique. Consider, for example, two identically located buildings (of different target types), each with EMDs fixed by the MAE for a structural kill. If one also inputs an effective kill radius based on equipment damage for one of the buildings and the kill radius for personnel with the other, the Monte Carlo results will include estimates of the structural damage, as well as of the materiel damage and personnel losses resulting from a structural hit.

## II. INPUT PROCEDURES

Up to seven basic types of cards can be used in operating AIDA, although only three are required. Four types describe the target and attack characteristics and three others are used to control operations. For each basic card the type is identified in the first four columns (left-adjusted):

TGT	target data card
ATT	attacker data card
ATT2	alternate attacker data card (optional)
EMD	weapon card (optional)
CONT	control card (optional)
REDO	controls sequential cases (optional)
END	last card

For two of these basic card types--ATT2 and EMD--supplementary cards are used. For the ATT2 card a following card (with additional data, as will be described) is mandatory. One to three supplementary cards may be employed with the EMD card, as will be described.

There may be as many as 250 TGT cards, 50 ATT or ATT2 cards, and 10 EMD cards. For a given case there may be at most 1 CONT card. The order of the cards is immaterial\* except that a REDO card or an END card must be used to signify the completion of input for a given case. The targets and attackers are numbered, internally, in the sequence in which their descriptions are read in; each target may also have an alphanumeric designator (e.g., building number). A detailed description of how data are to be entered on each type of card is presented in Appendix A. The input data are printed for each case; Table 1 outlines the output options for the results. The program will function in its simplest form with only TGT cards, ATT or ATT2 cards, and an END card.

---

\*The only exception is that if the special "target type #21" cards are used in conjunction with the special hit-density grid feature, one of these cards must be the last target card.

Table 1

OUTPUT CONTROL

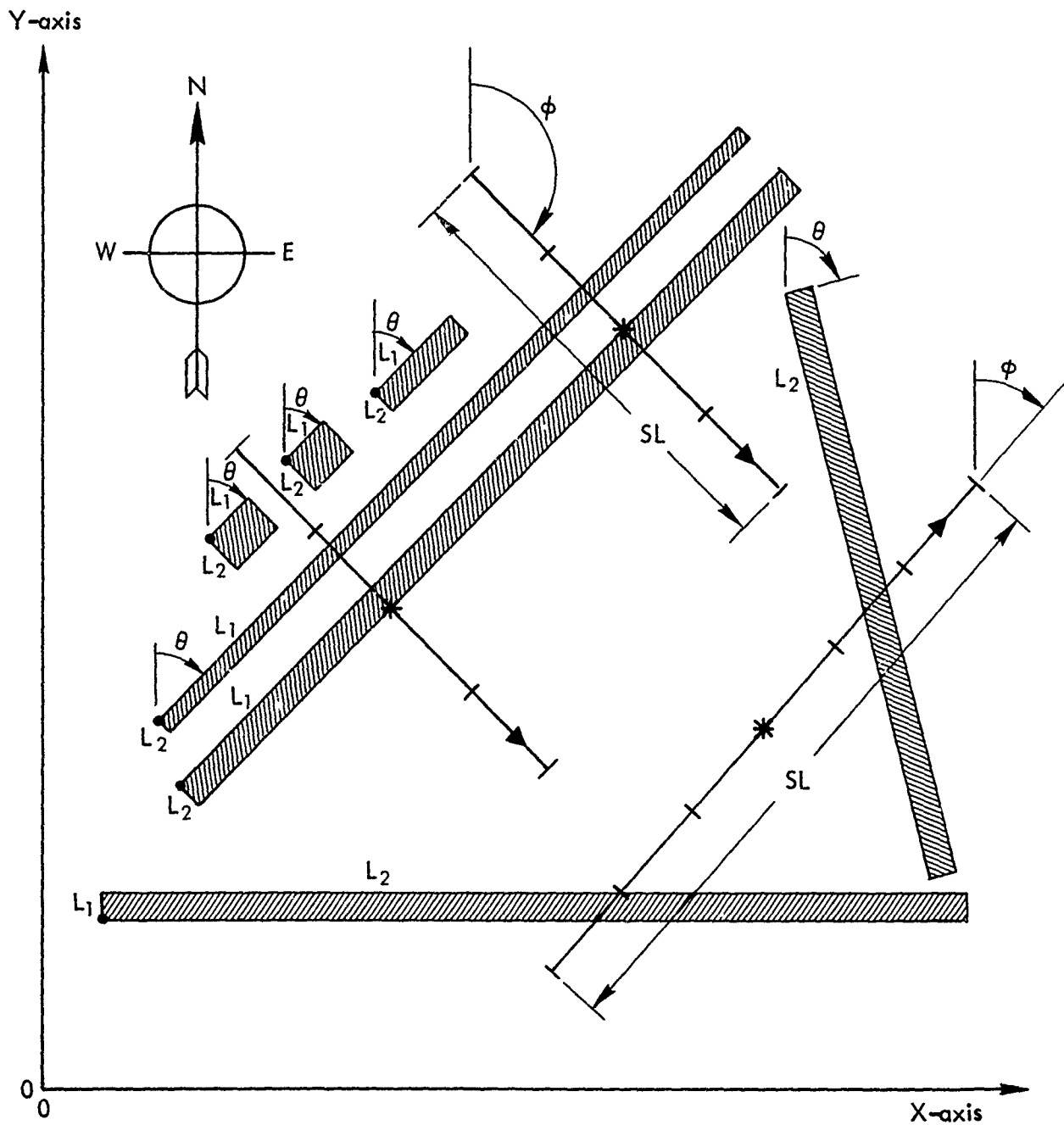
NPRINT <sup>a</sup> Control Value	For Each Trial					Multiple Trial Statistics
	All Impact Points	All Hits (and Target Corners)	Stored Hit Data	Hit Summary	Runway Results	
-2	X	X	X	X	X	X
-1		X	X	X	X	X
0			X	X	X	X
1				X	X	X
2					X	X
3						X
4					X <sup>b</sup>	X
5				X <sup>c</sup>		X

<sup>a</sup>Enter in Columns 23, 24 of CONT card.

<sup>b</sup>Compact listing of hits and required repairs for runways/taxiways.

<sup>c</sup>Compact listing of hits on each target.

A control (CONT) card will be required if advantage is to be taken of more than the most basic of AIDA's features; without this card AIDA examines only one Monte Carlo sample of the attack and provides the actual numbers of hits on all targets and the stored hit locations for specified targets. Specifically, a CONT card is needed if (1) more than one trial is required, (2) an alternative output mode is desired, (3) a different mode of operation is desired (e.g., the expected-value mode), or (4) the runway availability features are to be exercised. As explained more fully on pp. 38-39, this card is used to specify the number of trials, the mode of operations, the output formats, the dimensions of the minimum runway surface, whether or not the user wants the minimum repair requirement assessed, and the distances that the "minimum runway rectangle" will be shifted laterally and longitudinally in testing for an available area.



TARGETS

- Reference (westernmost) corner
- L<sub>1</sub> Northeasterly heading boundary
- L<sub>2</sub> Southeasterly heading boundary
- θ Orientation angle

ATTACK PASSES

- \* DMPI
- φ Attack heading
- SL Stick length
- †† Nominal bomb impacts

Fig. 1—Target and attack layout for AIDA

Several special features should be noted with respect to the EMD cards. To begin with, these cards are optional; if the EMD card is not used for one or more weapon types, these weapons are assumed to be point-impact weapons that must impact within the target outline to be counted as "hits." If more of the features are to be used for a particular type of weapon, an EMD card is input with the number of the weapon entered in Columns 11 and 12. If weapon reliability is less than unity, a 1 is placed in Column 6 of the EMD card and the reliability is entered in Columns 7 to 12 of a special card placed immediately following the EMD card. If the user wants to consider 20 types of targets, rather than only 10, a 1 is placed in Column 5, and data for the second set of 10 target types are entered on a supplementary card (2 cards if a 1 also appears in Column 6). The other entries for the EMD card and the special following cards differ, depending on whether it is a CBU-type munition or a point-impact weapon.

CBU munitions are denoted by a negative number in Columns 13 to 18 on the EMD card. The absolute value of that number is taken as the CBU pattern dimension along the flight direction; pattern width is specified in Columns 19 to 24. If the user wants to associate a probability of kill with CBU coverage on some or all of the target types, that fact is also denoted by the integer 1 in Column 6 of the EMD card. When this is done the program interprets entries on the following card (also used for weapon reliability) as  $p_k$  for that weapon against the desired target types.\*

Other special features are available for use with point-impact weapons. If a weapon can effectively damage a target when it actually falls near but outside the target outline, the EMD for a "hit" can be entered in Columns 13 through 72 (in 10 fields of 6 columns) of the EMD card (and auxiliary card) for the 10 (or 20) target types. The appropriate entry in many instances would be the radius of a circle whose area is equal to the MAE (as presented in JMEM manuals) for the corresponding target-weapon combination. In the case of hits on runways or taxiways, however, the appropriate entry is crater radius; when AIDA

---

\* In ten 6-column fields from Columns 13 to 72.



checks for the availability of a minimum runway, each reliable impact is assumed to have a crater radius equal to EMD.

For each "hit" with a point-impact weapon an estimate is made of the fraction of the target area that is covered by a circle of radius EMD.\* The user may also select one of the additional options (see Fig. 2); using the supplementary card he either may input a  $p_k$  for a particular weapon-target type or specify a radius different than EMD for defining another "effects" circle to be used in computing an additional coverage measure. In either case ( $p_k$  or radius) the value<sup>†</sup> is entered on the supplementary card in the field corresponding to the appropriate target type.

A REDO card terminates the input for one case and initiates a new case. If no additional entries are made on the REDO card, the targets and attacks for the next case will include all those entered in the preceding case. If some, but not all, of the preceding targets and/or attacks are to be included, the number of targets that are to be retained is entered in Columns 7 to 12 and the number of attacks to be retained is entered in Columns 13 to 18. The numbers to be retained are selected from the beginning of the ordered lists generated in the input process. If a negative entry (e.g., -1) is made in either or both of the target or attack fields, none of the prior targets and/or attacks will be considered in the new case.

After specifying which--if any--of the prior inputs are to be retained, additional targets and attacks may be added with appropriate TGT and ATT cards. If a 1 is entered in Column 24 of the REDO card, the target list and/or the attack and weapons lists will not be included in the input listing, *if* no change has been made in those lists from the prior use.

Control conditions will remain as in the preceding case unless a new CONT card is entered with appropriate data. An END card must be used, at the end of the last case, to complete the card deck.

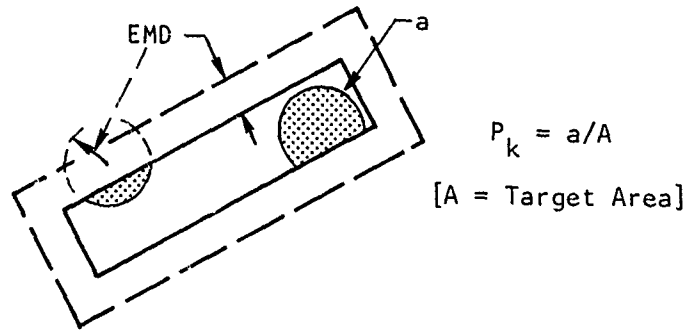
---

\* As noted earlier, the cumulative coverage of a target is given by 1 minus the product of the probability of noncoverage to all hits; this is equivalent to saying that in estimating the target area covered by a particular weapon the effect of prior hits is neglected.

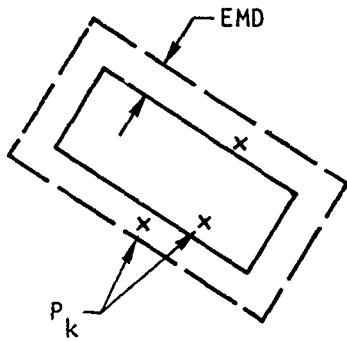
† A value greater than 1 is interpreted as a radius.

NOMINAL ESTIMATE

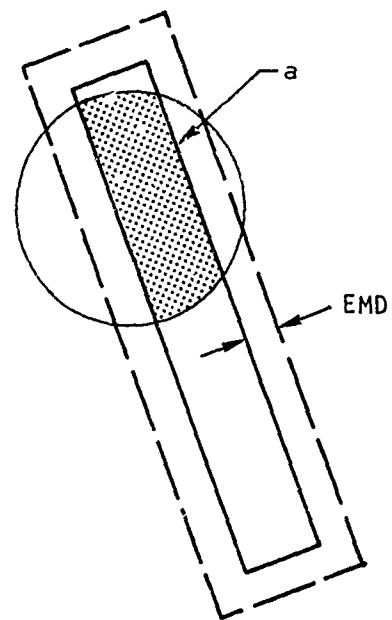
Radius = EMD



ALTERNATIVE I  
 $P_k$  input directly



ALTERNATIVE II  
Radius specified



CUMULATIVE COVERAGE

$$FC = 1 - \prod_{i = \text{all hits}} (1 - P_{k_i})$$

Fig.2 — Point-impact weapon coverage estimation options

### III. PROGRAM NOTES

AIDA is written in FORTRAN IV and first operated on the Honeywell 6060 used with Hq USAF's Worldwide Military Command and Control System (WWMCCS); subsequently, it has been run on the IBM 360/370 series. AIDA is easily transferable to other computers because there are no system-unique features employed.\* The program is composed of approximately 1950 card images organized into a MAIN routine and 17 subroutines. As presently dimensioned, AIDA operation requires 36K words of core. The organization of the various subroutines and their functions are indicated in Fig. 3; this structural outline of AIDA should also prove useful for anyone who wants to overlay the program so as to reduce core storage requirements. Potential users will find that the source code is moderately well annotated with comment cards. A full listing of the definitions for the key variables and arrays will be found in Appendix B.

AIDA's CPU requirements obviously can vary widely depending on the complexity of the problem (i.e., numbers of targets and attacks and use of the special runway features) and upon the required number of replications. For problems involving about 150 targets, 750 bombs, and 20 replications of the attack, computing times have run between 1-1/2 to 2 minutes on the Honeywell machine. To limit these requirements the process used to test for hits with point-impact weapons has been designed to reduce the TARGETS X BOMBS dimensionality problem somewhat, by ordering targets in a particular manner and by only checking those targets "near" each bomb impact point. Nevertheless there is a very substantial amount of checking and computation required.

The computation problem is especially severe when CBU munitions are used. To estimate the joint coverage of all CBU patterns, the targets each have a uniform grid of points superimposed, and each of those points is checked to see if it lies within one or more of the patterns.

---

\* Minor format changes may be required in a few instances; name changes will probably be necessary for the random number generator.

MAIN controls program operations and initializes numerous arrays

Reads, organizes, and lists input data  
Performs bomb trajectory calculations as required

Computes and stores target corner locations

Computes and prints hit-density grid on demand

Computes and prints expected number of hits for point-impact weapons

Orders targets by location of westernmost corner  
Groups targets into zones

Determines actual MPI and weapon-impact points

Checks for hits with point impact weapons

Checks for coverage by CBU-type weapons

Controls runway tests

Checks for runway minimums and determines minimum repairs  
Plots impact points on/near runways and main taxiways

Prints results from individual trials

Computes and prints the statistical results for multiple trials

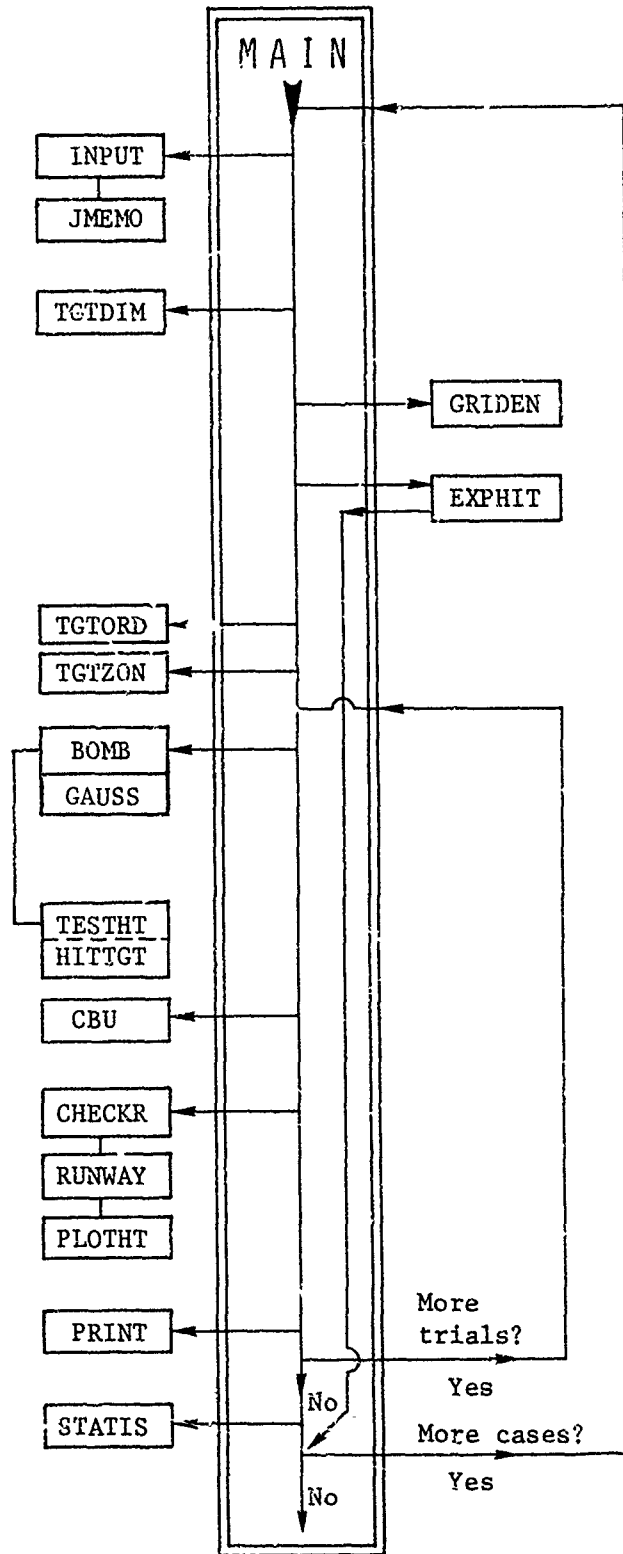


Fig.3— Organization and function of the AIDA subroutines

The fraction of the target grid points that is covered by one or more CBU patterns is used as the estimate of fractional coverage. For small targets only a  $4 \times 4$  grid is used, but for the very large targets (dimensions in excess of  $1000 \times 1000$ ) a  $16 \times 16$  grid is used;\* quite obviously the computational requirements are affected by the presence of large targets and the use of CBUs, but not, fortunately, in direct proportion to the number of grid points.

The procedure used for checking the availability of an adequate section of runway also involves substantial processing. A rectangle of the required dimensions is positioned first at a corner of the runway and tested to see if there is a hit within the rectangle. If there is, the rectangle is moved  $5 \text{ ft}^\dagger$  laterally and rechecked; this is repeated across the runway or until an open area has been found. If none is found the rectangle is shifted  $250 \text{ ft}^\dagger$  along the runway and the process is repeated. If the dimensions of the minimum usable runway section are small, or if the runway is large and/or there are many hits, a very substantial amount of processing can be required. These requirements are further increased when an assessment of the minimum repair requirements is requested. Furthermore, if there are other runways, or taxiways suitable for emergency flight operations (signified by entry as a target type #1), the entire process must be repeated for each surface (for a maximum of 5). Whenever the problem can be effectively reduced to one of establishing the availability of the minimum clear width (e.g., when a 9000 ft runway is being cut at the center to deny a 6000 ft clear length), the minimum clear length should be set equal to the runway length to avoid going through the lateral shift sequence unnecessarily (11 times in the example).

---

\* It will be noted that hits may not be recorded when a target is very large and the CBU pattern so small, proportionately, that the pattern can fall between grid points.

<sup>†</sup> These are the default values; the user may specify a different value.

#### IV. SAMPLE PROBLEM

The layout of Base XYZ is shown in Fig. 4. This base consists of a 150 ft × 8000 ft main runway, several taxiways, a parking ramp, eight support facilities, and a housing area. To examine the effectiveness of bombing attacks against this target complex with AIDA, one first must describe the target elements and the attack in a common coordinate system. Targets are defined by their westernmost corner, their size, and their orientation; the attack heading and the desired mean point of impact fix the attacks. For this illustration four medium bombers will each drop 25 bombs in an effort to cut the runway at two points; two others are targeted on the operations building near the main taxiway, and one will aim at the electronics shop. In addition, one fighter-bomber is assigned to dive-bomb each of the main aircraft maintenance buildings, B1 and B2, and one will drop a stick of 5 CBUs on the housing area.

#### INPUT

Figures 5a and 5b reproduce the card images needed to describe this sample problem and to control four distinctly different assessments. These assessments are designated as "cases" and illustrate the following:

- Case 1. Hit statistics for all targets, using 5 trials.
- Case 2. Detailed results of a single trial.
- Case 3. Results using the expected-value mode, with a sample hit-density grid.
- Case 4. Summary results of the runway attack, using 25 trials.

For clarity, the control (CONT) card, the target (TGT) cards, the weapon (EMD) cards, and the attack (ATT and ATT2) cards are listed in order; the ordering of cards for a given case is actually immaterial except that the EMD card pairs and the ATT2 card pairs must be together. A REDO card or an END card defines the end of the input data for a case.

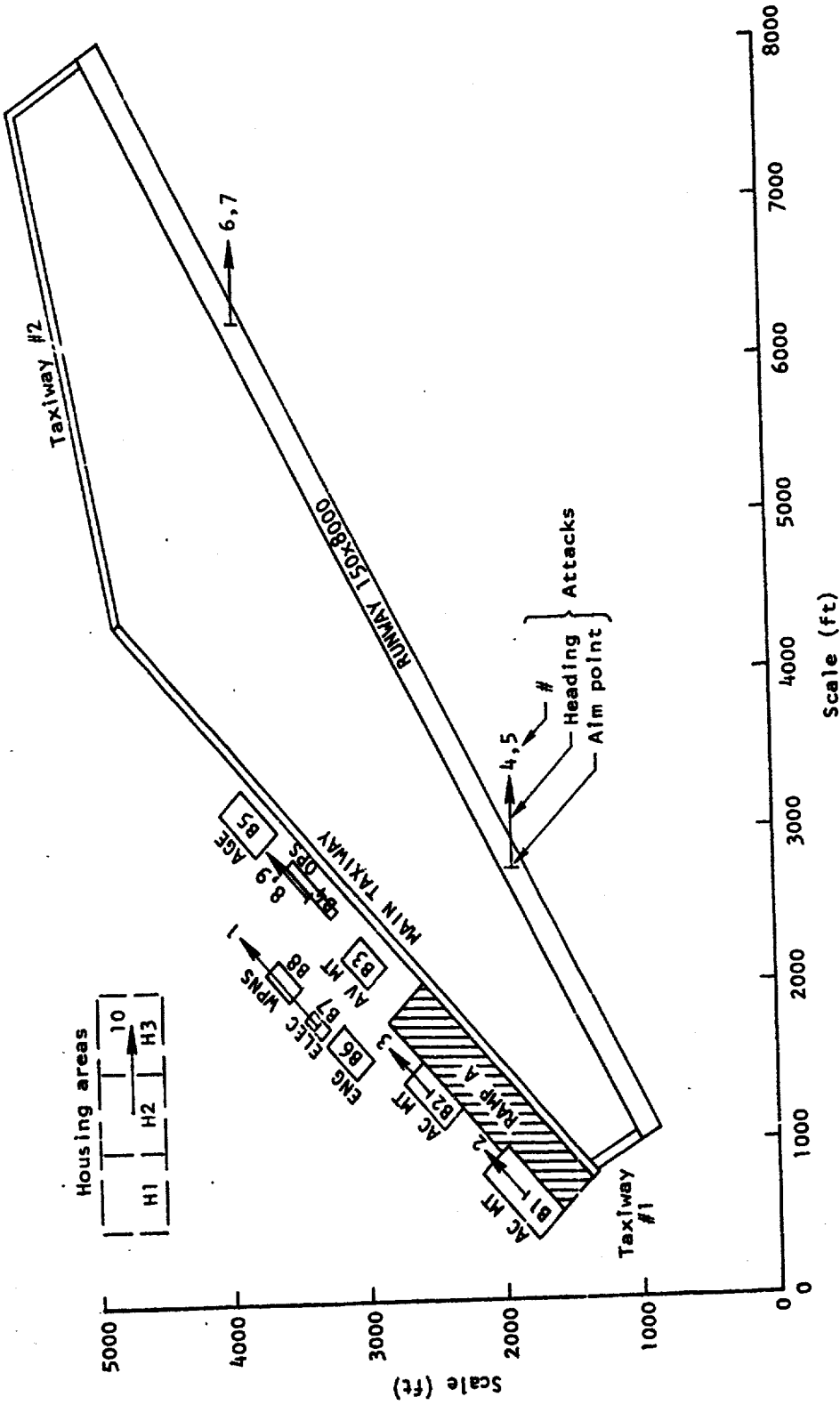


Fig. 4 — Base XYZ

Monte Carlo mode with fixed seed for random number generator  
Number of trials  
Statistical summary only

CONT			BASE XYZ		
TGT	1000 1000 8000 150 60' 1		FUNWAY		
TGT	77 1350 5000 50 45 1		MAIN TWY		
TGT	540 1610 1750 175 45 3		RAMP A		
TGT	400 1325 60 375 60 2		TXNY #1		
TGT	4400 4800 1300 50 78 2		TXNY #2		Optional descriptors
TGT	400 1750 600 250 45 4 1		AC RT B1		
TGT	1100 2450 400 200 45 4 1		AC RT B2		
TGT	2000 3000 250 150 45 5		AV RT B3		
TGT	2450 3220 500 60 45 4		OPS B4		
TGT	2450 3800 400 250 45 5		AGE B5		
TGT	1425 3075 300 175 45 4		ENS B6		
TGT	1680 1325 150 80 45 5		Stores hit locations ELEC B7		
TGT	1930 3575 225 100 45 5		WPNS BR		
TGT	500 4500 500 500 0 10		H 1		Housing areas
TGT	1000 4500 500 500 0 10		H 2		
TGT	1500 4500 500 500 0 10		H 3		
E*0 1	1 22 22 22 40 40				
END	1 28 28 28 50 50				
END 1	1 500 250				

TARGET CARDS

WEAPON CARDS

Number of trials  
 Target coordinates  
 Target dimensions  
 Target orientation  
 Target type  
 Stores hit locations  
 EMD--Hits are recorded for impacts within EMD of the target  
 Weapon reliability  
 Damage to type #4 and #5 targets is assessed with these radii, given a hit  
 Weapon type  
 Signifies CBU weapon  
 Pattern size, 300 x 250  
 PR for type #10 targets is assessed assuming 25% chance of kill within area covered

Fig. 5a — Input control card deck, Case 1



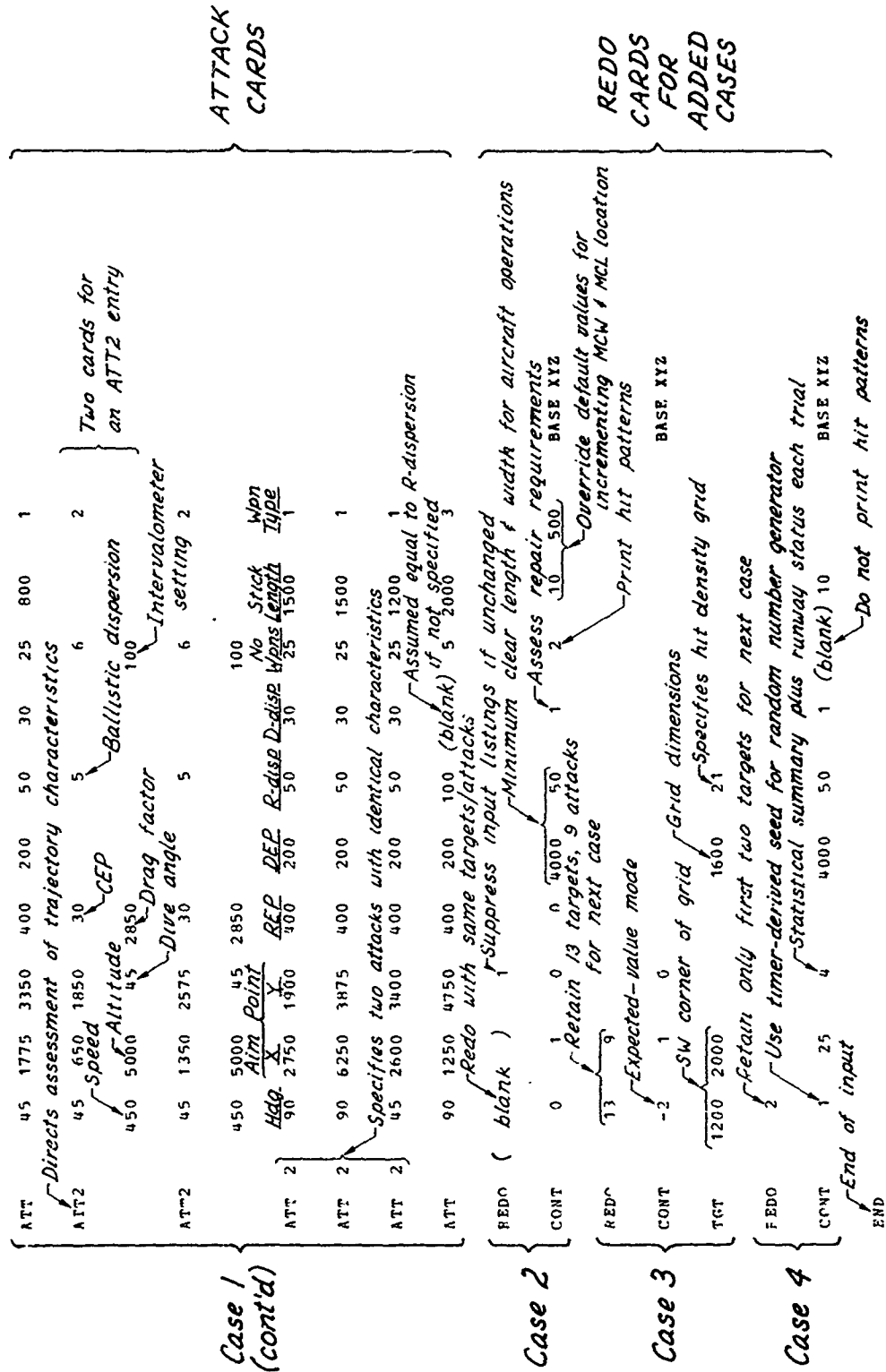


Fig. 5b — Input control card deck, Cases 1-4 (cont'd)

A careful review of the various annotations in Figs. 5a and 5b along with a reading of Appendix A, hopefully will lead to a full and rapid understanding of the various input requirements and optional features provided by AIDA. As can be seen in Case 1, the first card directs that the Monte Carlo mode be used and that a statistical summary of five replications of this attack be printed; no assessment of runway availability is requested for this case. The specifications for the targets are straightforward. The weapon data are somewhat more complex; two cards are used in describing each weapon's effectiveness against the 10 types of target (four cards would be required for 11 to 20 target types). The first card lists the EMD for 10 target types and the second card lists weapon reliability and the optional effectiveness descriptor (an alternate radius, or a  $p_k$ ). In this example weapon type #1 has an effective (crater) radius against paved surfaces (target types #1, #2, and #3) of 22 ft; against target types #4 and #5 a hit will be assessed if the impact is within 40 ft of these structures. The fractional damage on these structures, given a hit, is to be based on a 50 ft effects radius for target type #4, and a 75 ft radius for target type #5. The third weapon type is a CBU and the special input procedures are illustrated.

The 10 attacks are listed last. The first (and fourth through tenth) attack uses the simpler ATT specification; the REP, DEP, dispersion, and stick length are given directly in feet, measured in the ground plane. For the second and third attacks--the fighter-bombers directed to attack the main maintenance hangars B1 and B2--these factors are to be computed, based on the aircraft's attack characteristics; both will release their last bomb at a 5000 ft altitude in a 45 deg dive at 450 kn with an intervalometer setting of 100 ms. Aiming accuracy and dispersion in the plane normal to the trajectory are 30 mils and 5 mils, respectively. For three of the attacks, two aircraft will attempt to fly the identical path, so only one card is needed to describe each pair.

The remaining eight cards are all that are needed to define and control three additional cases. Case 2 simply calls for a single Monte Carlo attack, but with full printout and with an examination of the

availability of an undamaged 50 ft × 4000 ft section of either the runway or main taxiway for aircraft operations.

Case 3 requests an estimate of the expected numbers of hits using the expected-value mode; since this mode will not function for CBU-type munitions, only the first nine attacks are retained (the housing area targets for the tenth attack are also dropped). The target card that is added in this case directs the additional computation of a 1600 ft × 1600 ft hit-density grid, located as noted.

Case 4 focuses on the availability of a minimum surface for aircraft operations. Only the runway and main taxiway are retained as targets; however, all attacks are to be considered, since any crater must be taken into account, whether it represents a hit on the intended target, or collateral damage from some different attack. Twenty-five attack trials are to be run with the Monte Carlo mode, repair requirements are to be assessed, and the trial-to-trial runway results are to be printed along with the statistical summary for the 25 trials.

#### OUTPUT

The very first output (Fig. 6), presented even before the main summary of the input data, lists the input/output for any trajectory calculations that were required; in this case this was necessary for attacks 2 and 3. The basic input listing follows; as will be noted, the factors determined with the trajectory calculations are listed in their normal location in this format. It will also be noted that the targets and the attacks were assigned numbers in the order in which they were located in the input deck; the pairs of identical attacks each were assigned two numbers.

The target damage statistics for five Monte Carlo replications of the attack in Case 1 are shown in Fig. 7. The various annotations will help to clarify the nature of the various statistics that are provided. As can be noted in conjunction with the type #10 targets, "hits" are not assessed with CBU weapons; the results are in terms of coverage and fractional kill (PK). Perhaps the most significant--and typical--observation that might be drawn from these sample statistics is the very substantial uncertainty that must be attached to any measures of

Inputs and trajectory data  
for dive bombing attacks  
on buildings B1 and B2

ATT NO	HQS SPED	X-WFI ALTITUDE	(-MFI DIVE	CFP (REF) TIME	(DEP) TYPE	DISP WIND	TD/HP	T	TD
2	450.	f=0. 5000.	1850. 45.	10. 1200.	0. 0.	5. 0.	0.0	0.100	0.0
			TF 7.83 SEC	S	6795.	FEET	IMPACT ANGLE 55.85 DEG	(FUZZING ANGLE 0.0 DEG)	
1	450.	1350. 5000.	2575. 45.	10. 1850.	0. 0.	5. 0.	0.0	0.100	0.0
			TF 7.83 SEC	S	6795.	FEET	IMPACT ANGLE 55.85 DEG	(FUZZING ANGLE 0.0 DEG)	

Intervalometer  
setting

\*\*\*\*\*  
AN AIRBASE DAMAGE ASSESSMENT MODEL  
DEVELOPED BY THE TANSTEIN OFFICE OF THE RAND CORPORATION  
NO OF TRIALS 5 (MINIMUM) 0 (250) MCW 0 (5) MIN REPAIR 0 PLOT HITS 0 TEST 0  
\*\*\*\*\*

Statistical summary only

\*\*\*\*\* BASE COMPLEX NAME - ENIE XYZ \*\*\*\*\*

NUMBER	X-TYP	TARGET DATA Y-NUM	LINE	SF	LINE	ANGLE	TST	TYPE	STOPE	BLDG NO
1	1000.	1000.	8000.	17.0	60.	1.				PUNWAY
2	775.	1350.	5000.	5.0	45.	1.				WATER TMT
3	540.	1610.	1750.	17.5	45.	3.				WATER TMT
4	800.	1327.	330.	17.5	70.	2.				WATER TMT
5	400.	4900.	3300.	27.0	45.	4.				WATER TMT
6	1100.	2750.	400.	27.0	45.	4.				AC MT B1
7	1100.	2750.	400.	27.0	45.	4.				AC MT B2
8	2400.	2300.	250.	17.0	45.	4.				AV HT B3
9	2400.	2300.	250.	17.0	45.	4.				AV HT B4
10	2400.	2300.	250.	17.0	45.	4.				OPS B8
11	1625.	3075.	300.	25.0	45.	5.				AGE B5
12	1680.	3125.	300.	17.5	45.	4.				ENG B6
13	1730.	3575.	225.	10.0	45.	5.				ELEC B7
14	500.	4500.	500.	50.0	0.	10.				WPS B8
15	1000.	5500.	500.	50.0	0.	10.				H 1
16	1500.	4500.	500.	50.0	0.	10.				H 2
17	1500.	4500.	500.	50.0	0.	10.				H 3

NUMBER	HQS	ATTACK DATA X-DIFF	Y-DIFF	RFI	DEP	P-DISP	D-DISP	NO WPS	LENGTH	WPN	TYPE	ARRIVAL
1	45.	1775.	1350.	400.	200.	50.	30.	25.	800.	1.	1-000	1-000
2	45.	1650.	1650.	350.	117.	28.	23.	6.	86.	2.	1-000	1-000
3	45.	1750.	1900.	200.	200.	50.	30.	25.	1500.	1.	1-000	1-000
4	40.	2750.	1900.	400.	200.	50.	30.	25.	1500.	1.	1-000	1-000
5	40.	2750.	1900.	400.	200.	50.	30.	25.	1500.	1.	1-000	1-000
6	40.	2750.	1900.	400.	200.	50.	30.	25.	1500.	1.	1-000	1-000
7	40.	2750.	1900.	400.	200.	50.	30.	25.	1500.	1.	1-000	1-000
8	40.	2750.	1900.	400.	200.	50.	30.	25.	1500.	1.	1-000	1-000
9	45.	2600.	3400.	400.	200.	100.	100.	5.	2000.	1.	1-000	1-000
10	40.	1250.	4750.	400.	200.	100.	100.	5.	2000.	1.	1-000	1-000

As computed with the JMEM method

CBU; see below

MISS DISTANCES ALLOWED FOR EFFECTIVE HITS

WPN TYPE	1	2	3	4	5	6	7	8	9	10
1	0.950	22.	22.	22.	40.	40.	0.	0.	0.	0.
2	0.950	28.	28.	28.	50.	50.	0.	0.	0.	0.
3	0.900	250.	250.	250.	60.000	90.000	0.	0.	0.	0.
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.250

Crater radii vs pavements

Damage radii given a hit

CBU

Fig. 6 — Target, attack, and trajectory data, Cases 1-4



the "average level of damage": as can be seen, the standard deviation is frequently as large, or larger, than the mean.

Case 2 called for a full printout of one trial, but with the input data suppressed. These results are shown in Figs. 8 through 10. The first results presented are the hit patterns on the runway (Fig. 8) and main taxiway (Fig. 9) as well as statements as to their status. As will be noted, all hits within an EMD of the target are included, since a crater will affect the surface up to a distance equal to the crater radius. In the case of the taxiway against which no attack was specifically planned, it is apparent that either an attack aimed at the operations building or one of those aimed at buildings B1 or B2 was wide and to the right of the aim point. A careful study of the target hit summary (Fig. 10) indicates that it was one of the attacks on the operations building, since the hits are with type #1 weapons. Using the EMD of 40 ft for structural damage and a 75 ft radius for the contents of type #5 buildings, the results show that this attack will destroy an estimated 15.7 percent of the structure and damage contents on 57.5 percent of the floor space of the avionics maintenance facility (B3); on the other hand, the AGE facility (B5), which is similarly located with respect to the intended aim point near the operations building, was missed entirely as was the intended target. One of the main aircraft maintenance hangers (B2) was hit. Other dive-bombing attacks were both wide and to the right, placing bombs on the main parking ramp. For the housing areas, there was no damage to the exposed targets in the three areas in this particular trial. The actual hit locations on building B2 and on the operating surfaces are shown at the bottom of Fig. 10. Of the four targets for which hits were to be retained (#6, #7, #11, and #13), only #7 was hit in this case.

Case 3 used the expected-value mode to provide estimates of the expected numbers of hits and to compute a hit-density grid. The latter is controlled by specifying a "type #21" target; as will be noted in Fig. 11, the grid dimension is printed in the position normally reserved for target angle. Any grids that have been specified are computed first and the sample requested in this case is shown in Fig. 12; as will be noted, the results are presented in terms of the expected numbers of

```

*****
***** AN AIRBASE DAMAGE ASSESSMENT MODEL
DEVELOPED BY THE RAMSTEIN OFFICE OF THE RAND CORPORATION
*****
NC OF TRIALS 1 | PRINT 0
MODE 0 MCL 1000 | 500 MCM 50 | 10 | MIN REPAIR 1 | PLOT HITS 2 | TEST 0
*****
***** BASE COMPLEX NAME - BASE XYZ *****
***** TRIAL # 1 *****

```

-Plot any hits on potential operating surfaces  
 -Record minimum repairs required for minimum operating surface size  
 -Minimum operating surface size  
 -Calls for full listing of trial data (1 trial)

If either of these craters were repaired, a minimum operating surface would be available

Note: Badly distorted scale is necessitated by paper size

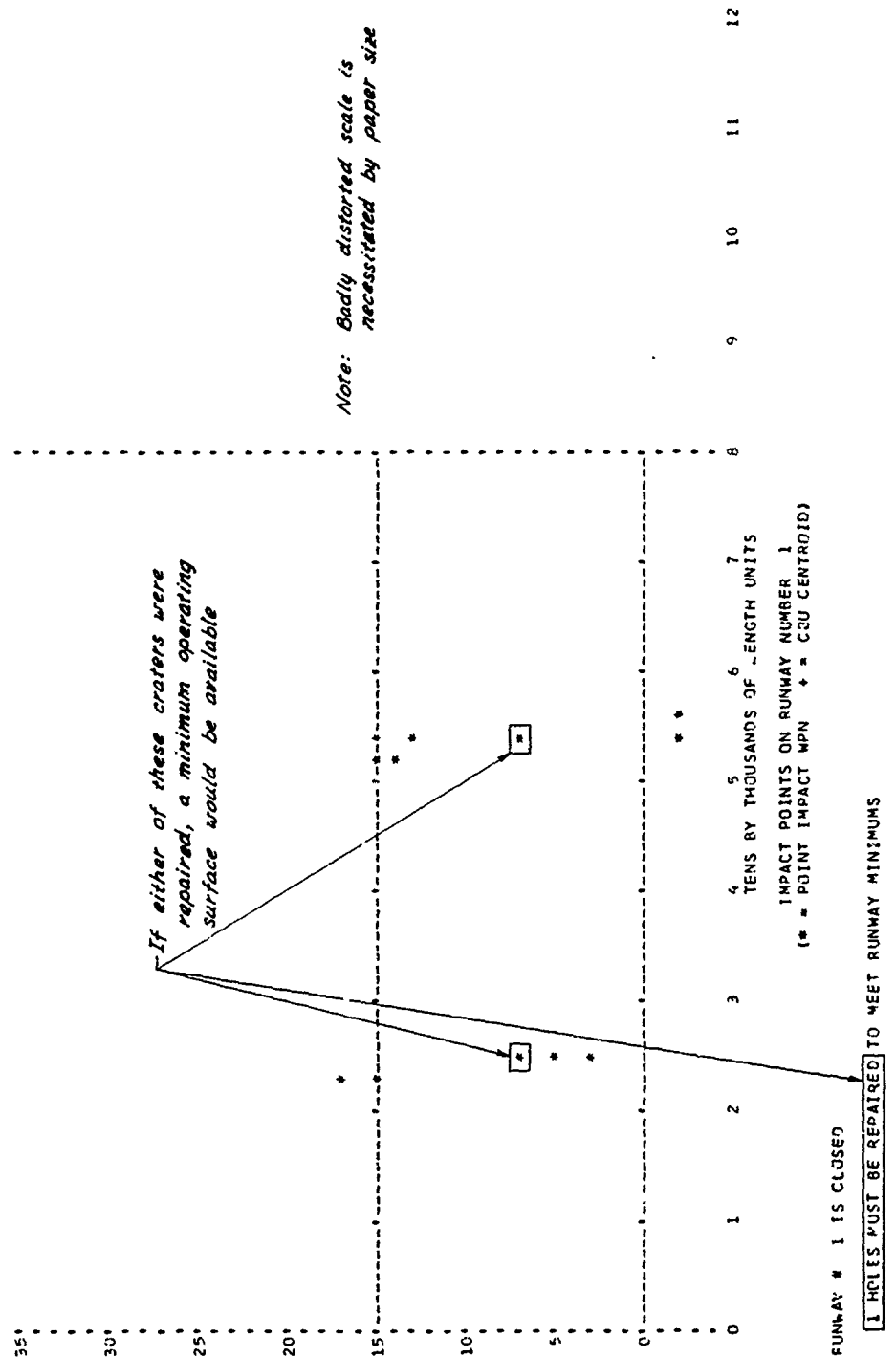


Fig. 8 — Runway hit patterns, Case 2

*Although no attack was targeted against this main taxiway, four attacks (#2,3,8 and 9) are targeted nearby and at least one obviously was wide and to the right on this trial. Since, as will be seen, the impacts are for type #1 weapons, it was either attack #8 or #9 that hit this taxiway.*

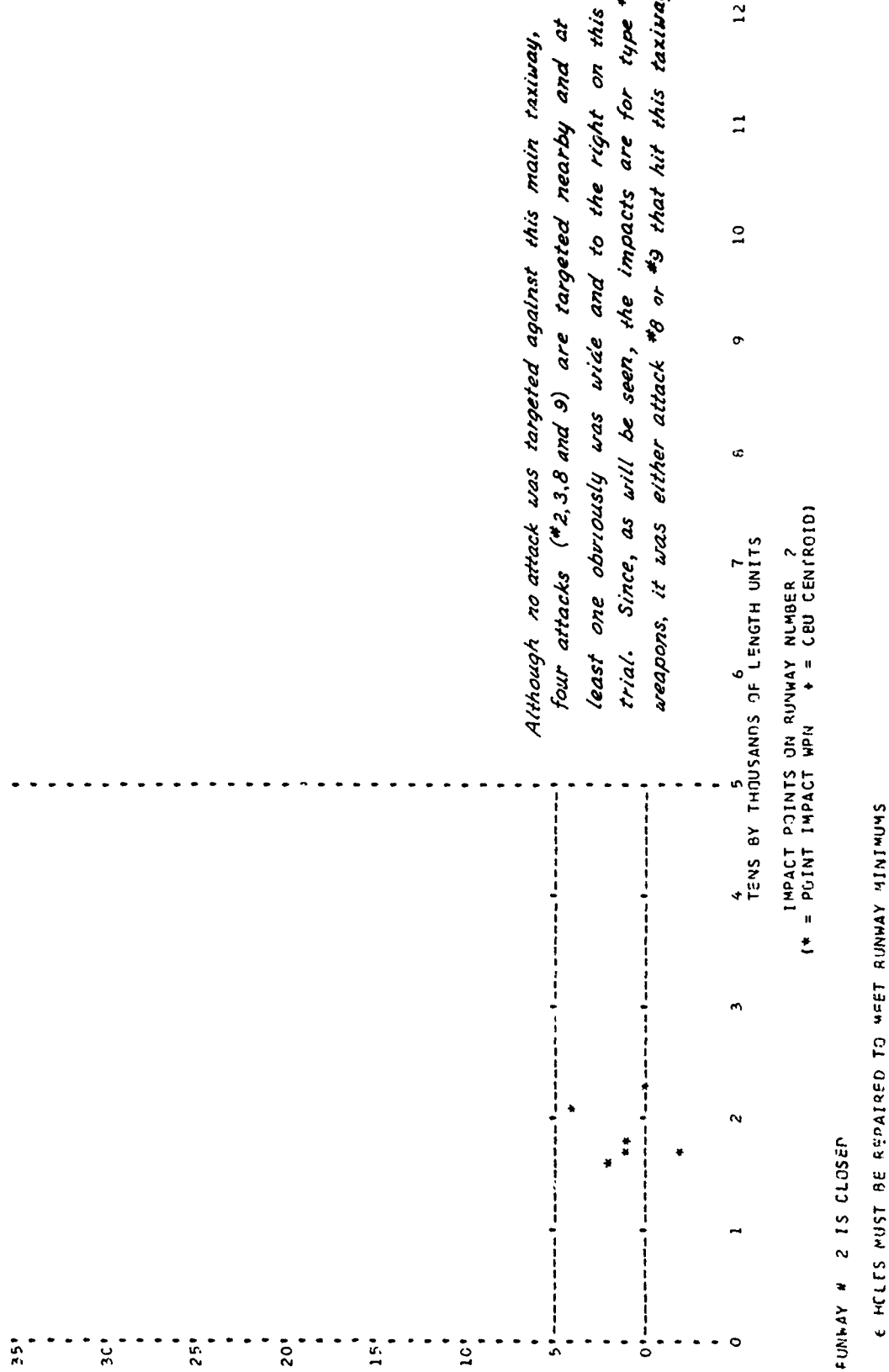


Fig. 9 — Taxiway hit patterns, Case 2



TARGET HIT SUMMARY

TGT P.O.	A.U. HIT%	CHU COVERAGE	BOMBS EMP (OTHER PK)	CBU PK	BLDG NO.
1	13	0.0	0.0	0.0	RUNWAY
2	0.0	0.0	0.0	0.0	4A IN TMY
** TARGET TYPE # 2 **					
4	0	0.0	0.0	0.0	TMY #1
5	0	0.0	0.0	0.0	TMY #2
** TARGET TYPE # 3 **					
3	19	0.0	0.040	0.0	RAMP A
** TARGET TYPE # 4 **					
6	0	0.0	0.0	0.0	AC NT B1
7	4	0.0	0.093	0.0	AC NT B2
9	0	0.0	0.0	0.0	TPS B4
11	0	0.0	0.0	0.0	ENG RB
** TARGET TYPE # 5 **					
6	0	0.0	0.151	0.0	AV NT B3
10	0	0.0	0.373	0.0	AGE B5
12	0	0.0	0.0	0.0	ENG B8
13	0	0.0	0.0	0.0	WPS B8
** TARGET TYPE # 10 **					
14	0	0.0	0.0	0.0	H 1
15	0	0.0	0.0	0.0	H 2
16	0	0.0	0.0	0.0	H 3

See sketches (Fig 9) and location listing below

With 4 hits, 15.7% of the area of this building covered by damage effects circles of radius = EMD (a)

57.5% of the alternate damage effects radius (75 ft) by the alternate damage effects radius (75 ft)

HIT LOCATION AND WPN TYPE FOR SELECTED TARGETS

TARGET NUMBER	7	X-DIM	Y-DIM	WPN TYPE
1	1517.	2587.	2	
	1516.	2612.	2	
	1496.	2633.	2	
	1491.	2514.	1	
2	3060.	2201.	1	
	3040.	2202.	1	
	3270.	2182.	1	
	3213.	2193.	1	
	3275.	2209.	1	
	3269.	2174.	1	
	5523.	3601.	1	
	5565.	3641.	1	
	5703.	3720.	1	
	5764.	3667.	1	
	5697.	3697.	1	
	5834.	3600.	1	
3	6013.	3701.	1	
	1964.	2503.	1	
	2038.	2567.	1	
	2071.	2551.	1	
4	2118.	2638.	1	
	2304.	2865.	1	
	2500.	3012.	1	

Hits were to be saved on targets #6, #7, #11, and #13; only #7 was hit

Fig. 10 --- Target hit summary, Case 2

\*\*\*\*\*  
 \*\*\*\*\* AN AIRBASE DAMAGE ASSESSMENT MODEL  
 / DEVELOPED BY THE RAMSTEIN OFFICE OF THE RAND CORPORATION  
 \*\*\*\*\*  
 NC OF TRIALS 1 PRINT 0 [MODE -2] [CL 0 (250) [CM 0 (5) MIN REPAIR 0 PLOT HITS 0 TEST 0  
 \*\*\*\*\*  
 Directs use of the Expected-Value Mode  
 \*\*\*\*\* BASE COMPLEX NAME - PASE XYZ \*\*\*\*\*

NUMBER	X-DIM	TARGET DATA		SE LIME	ANGLE	TGT TYPE	STORE	ELDG NO	WPN	DEF	P-DISP	D-DISP	NO WPNS	LENGTH	WPN TYPE	ARRIVAL
		Y-DIM	Z-DIM													
1	1000.	1000.	8000.	150.	60.	1.	0.	0.	0.	1600.	50.	30.	25.	800.	1.	1.000
2	775.	1350.	5000.	50.	45.	1.	0.	0.	0.	1600.	28.	23.	6.	86.	2.	1.000
3	540.	1610.	1750.	375.	60.	2.	0.	0.	0.	1600.	28.	23.	6.	86.	2.	1.000
4	600.	1325.	50.	375.	60.	2.	0.	0.	0.	1600.	50.	30.	25.	1500.	1.	1.000
5	4400.	4800.	3300.	50.	78.	2.	0.	0.	0.	1600.	50.	30.	25.	1500.	1.	1.000
6	400.	750.	900.	250.	45.	4.	1.	1.	1.	1600.	50.	30.	25.	1500.	1.	1.000
7	1100.	2450.	400.	200.	45.	4.	1.	1.	1.	1600.	50.	30.	25.	1500.	1.	1.000
8	2000.	3000.	250.	150.	45.	5.	0.	0.	0.	1600.	50.	30.	25.	1500.	1.	1.000
9	2850.	3220.	500.	60.	45.	4.	0.	0.	0.	1600.	50.	30.	25.	1200.	1.	1.000
10	2850.	3600.	400.	250.	45.	5.	0.	0.	0.	1600.	50.	30.	25.	1200.	1.	1.000
11	1425.	4075.	300.	175.	45.	4.	1.	1.	1.	1600.	50.	30.	25.	1200.	1.	1.000
12	1480.	3325.	150.	80.	45.	5.	0.	0.	0.	1600.	50.	30.	25.	1200.	1.	1.000
13	1480.	3575.	225.	100.	45.	5.	0.	0.	0.	1600.	50.	30.	25.	1200.	1.	1.000
14	1200.	2000.	0.	0.	45.	5.	0.	0.	0.	1600.	50.	30.	25.	1200.	1.	1.000

Special target type #21 added:  
 requests a hit-density grid over  
 a 1600' x 1600' square with the  
 southwest corner at x=1200, y=2000

NUMBER	HDS	ATTACK DATA		WPN	DEF	P-DISP	D-DISP	NO WPNS	LENGTH	WPN TYPE	ARRIVAL
		X-DIM	Y-DIM								
1	45.	1775.	3350.	400.	200.	50.	30.	25.	800.	1.	1.000
2	45.	650.	1850.	151.	117.	28.	23.	6.	86.	2.	1.000
3	45.	1350.	2575.	151.	117.	28.	23.	6.	86.	2.	1.000
4	90.	2750.	1900.	400.	200.	50.	30.	25.	1500.	1.	1.000
5	90.	2750.	1900.	400.	200.	50.	30.	25.	1500.	1.	1.000
6	90.	6250.	3875.	400.	200.	50.	30.	25.	1500.	1.	1.000
7	90.	6250.	3875.	400.	200.	50.	30.	25.	1500.	1.	1.000
8	45.	2600.	3400.	400.	200.	50.	30.	25.	1200.	1.	1.000
9	45.	2600.	3400.	400.	200.	50.	30.	25.	1200.	1.	1.000

Note that DU attack was dropped  
 since this mode will not function  
 if a CBU is in attack

MISS DISTANCES ALLOWED FOR EFFECTIVE HITS

WPN TYPE	WPN REL	TARGET TYPES														
		1	2	3	4	5	6	7	8	9	10					
1	0.950	22.	22.	22.	40.	40.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
2	0.950	28.	28.	28.	50.	50.	75.000	75.000	50.	50.	0.	0.	0.	0.	0.	0.
		0.0	0.0	0.0	60.000	60.000	90.000	90.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Fig. 11 -- Input data, Case 3

EXPECTED HIT DENSITY PER 10000 SQ FT

For different sized facilities, estimate the expected hits in proportion to their size

Y LOC	X - LOCATION																
	1200	1300	1400	1500	1600	1700	1800	1900	2000	2100	2200	2300	2400	2500	2600	2700	2800
3600	0.028	0.046	0.068	0.096	0.127	0.159	0.188	0.213	0.233	0.252	0.270	0.291	0.315	0.337	0.353	0.357	0.345
3500	0.042	0.065	0.093	0.126	0.160	0.192	0.221	0.245	0.265	0.284	0.305	0.328	0.351	0.369	0.377	0.368	0.342
3400	0.059	0.086	0.119	0.154	0.189	0.220	0.247	0.269	0.289	0.309	0.331	0.354	0.374	0.384	0.380	0.358	0.319
3300	0.076	0.107	0.142	0.178	0.211	0.239	0.263	0.283	0.303	0.324	0.346	0.366	0.378	0.378	0.361	0.327	0.279
3200	0.093	0.126	0.161	0.194	0.223	0.247	0.267	0.286	0.306	0.326	0.346	0.360	0.363	0.351	0.323	0.280	0.229
3100	0.108	0.142	0.175	0.205	0.228	0.246	0.262	0.279	0.298	0.317	0.331	0.337	0.330	0.308	0.272	0.226	0.177
3000	0.123	0.159	0.192	0.216	0.230	0.240	0.250	0.264	0.280	0.294	0.302	0.299	0.283	0.254	0.215	0.171	0.128
2900	0.144	0.187	0.221	0.237	0.237	0.232	0.233	0.241	0.253	0.261	0.262	0.251	0.229	0.197	0.160	0.122	0.088
2800	0.177	0.233	0.267	0.268	0.246	0.223	0.212	0.214	0.219	0.222	0.216	0.200	0.176	0.146	0.114	0.084	0.058
2700	0.220	0.286	0.314	0.294	0.248	0.208	0.187	0.183	0.184	0.181	0.171	0.154	0.131	0.106	0.081	0.059	0.041
2600	0.250	0.314	0.329	0.289	0.229	0.182	0.160	0.154	0.152	0.147	0.136	0.120	0.101	0.082	0.064	0.050	0.039
2500	0.243	0.293	0.291	0.246	0.190	0.152	0.136	0.131	0.130	0.125	0.116	0.104	0.091	0.078	0.068	0.059	0.053
2400	0.198	0.228	0.219	0.184	0.148	0.127	0.121	0.122	0.122	0.121	0.116	0.110	0.103	0.097	0.092	0.088	0.085
2300	0.140	0.155	0.149	0.133	0.121	0.117	0.121	0.127	0.133	0.136	0.138	0.138	0.138	0.138	0.138	0.137	0.135
2200	0.095	0.104	0.107	0.108	0.112	0.122	0.134	0.146	0.157	0.167	0.176	0.183	0.189	0.194	0.197	0.199	0.198
2100	0.072	0.086	0.090	0.102	0.117	0.134	0.152	0.170	0.187	0.204	0.219	0.232	0.243	0.252	0.259	0.262	0.261
2000	0.063	0.072	0.086	0.104	0.123	0.145	0.167	0.190	0.212	0.233	0.253	0.270	0.285	0.297	0.305	0.309	0.309

Southwest corner X=1200, Y=2000

CYCLES IN SPIDIN 16113

(Counts uses of EXP function)

Grid dimensions 1600' x 1600'

Fig. 12 — Expected hit density, Case 3

hits per 10,000 sq ft, or, in effect, within the walls of a 100 × 100 ft building centered at the grid position.

Figure 13 presents the expected hits for the regular target array. As a comparison will quickly indicate, the particular results for the single trial in the second case are substantially different from the average result that is to be expected. Furthermore, as comparisons with the statistical averages presented in Case 1 will confirm, a rather large number of trials are required if one is to obtain reasonably accurate estimates of a mean in problems of the sort examined here using Monte Carlo techniques. It is for reasons such as this that it was suggested earlier that the two modes can be used together beneficially so as to more efficiently obtain both a reasonably good estimate of the mean as well as useful evidence on the variability about that mean.

Case 4 examines the availability of a minimum operating surface for aircraft. Since this type of problem is rather complex and tends to demand relatively large numbers of trials, it will often be preferable to separate this examination from others, as we have here in Cases 1 and 4, to avoid unnecessarily processing data relating to attacks and targets that are not of interest. For this reason all targets but the two of interest have been dropped (see Fig. 14), and all attacks that could not have any effect should also be dropped; for this case we retained all the attacks, since none are extremely distant from both surfaces.

The first result in Case 4 is the trial-by-trial record (top of Fig. 15) of the total number of hits and minimum numbers of repairs required for both of the targets; there is no entry only if there are no hits. This record is especially useful if one wishes to examine the distribution of attack results in more detail than can be done using the overall statistical results themselves. The latter are presented at the bottom of Fig. 15. As can be seen by comparing these results for the taxiway with those in the previous case, even 25 trials is inadequate to assure a reasonable estimate of the mean (which is actually nearly twice that which has been obtained in this sample of 25 trials).

CYCLES 17731

Counts uses of EXP function

TARGET NO	HITS EXPECTED
** TARGET TYPE # 1 **	
1	15.454 RUNWAY
2	5.507 MAIN TWY
	20.962(0.998)
** TARGET TYPE # 2 **	
4	0.007 TWY #1
5	0.006 TWY #2
	0.014(0.007)
** TARGET TYPE # 3 **	
3	8.242 RAMP A
	8.242(1.000)
** TARGET TYPE # 4 **	
6	4.251 AC MT B1
7	3.850 AC MT B2
9	2.797 OPS B4
11	2.072 ENG B6
	12.970(0.944)
** TARGET TYPE # 5 **	
8	2.236 AV MT B3
10	3.623 AGE B5
12	0.836 ELEC B7
13	1.203 WPNS B8
	7.898(0.783)

Expected fraction of the facilities of this type with at least one hit

Total number of hits expected on targets of this type

Fig. 13 — Expected target damage, Case 3

```

*****
*****ALDA*****
*****      AN AIRBASE DAMAGE ASSESSMENT MODEL
*****      DEVELOPED BY THE PANSTEIN OFFICE OF THE RAND CORPORATION
*****
NO OF TTIALS 25 NFPINT 4 MODE 1 MCL 4000 ( 250) MCH 50 (10) MIN REPAIR 1 PLOT HITS 0 TEST 0
*****

```

Calls for compressed listing of runway status by trial

\*\*\*\*\* BASE COMPLEX NAME - BASE XYZ \*\*\*\*\*

NUMBER	TARGET DATA		X-DIM	Y-DIM	SF LTR	NE LTR	SF LTR	ANGLE	TGT TYPE	STORE	BLDG NO	NO NPNS	U-DISP	H-DISP	DEP	NFF	Y-DYPI	ATTACK DATA	LENGTH	MPN TYPE	ARRIVAL
	X-BRFF	Y-BRFF																			
1	1000.	1000.	775.	1350.	8000.	5000.	150.	60.	1.	0.	0.	25.	30.	50.	200.	400.	3350.	1775.	800.	1.	1.000
2	775.	1350.	1000.	1350.	8000.	5000.	50.	45.	1.	0.	PUNWAY	6.	23.	28.	117.	151.	1850.	650.	86.	2.	1.000
										C.	MAIN TWY	25.	30.	50.	200.	400.	1900.	2750.	1500.	1.	1.000
												25.	30.	50.	200.	400.	1900.	2750.	1500.	1.	1.000
												25.	30.	50.	200.	400.	3875.	6250.	1500.	1.	1.000
												25.	30.	50.	200.	400.	3875.	6250.	1500.	1.	1.000
												25.	30.	50.	200.	400.	3400.	2600.	1200.	1.	1.000
												25.	30.	50.	200.	400.	3400.	2600.	1200.	1.	1.000

Note that all targets other than the potential aircraft operating surfaces have been dropped as directed on the CONTROL card

MISS DISTANCES ALLOWED FOR EFFECTIVE HITS

MTR TYPE	WON PPL	TARGET TYPES																				
		1	2	3	4	5	6	7	8	9	10											
1	0.950	22.	22.	22.	40.	40.	40.	40.	40.	40.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.950	26.	28.	28.	50.	50.	50.	50.	50.	50.	75.000	75.000	75.000	75.000	75.000	75.000	75.000	75.000	75.000	75.000	75.000	75.000
		0.0	0.0	0.0	60.000	60.000	60.000	60.000	60.000	60.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000	90.000

Note: These different radii for type #1 and #2 weapons will be taken into account in assessing runway status. (However, it is unlikely that #2 weapons will actually be involved in this example as aim is too accurate.)

Fig. 14 - Input data, Case 4

**TARGET DAMAGE STATISTICS FOR 25 TRIALS**

TARGET NUMBER	PERCENT ATTACKS HIT	AVERAGE HITS PER ATTACK	STD. DEV. OF HITS	AVG. CRU SFD. DLY. COVERAGE	AVG. PCMB OTHER EMP. COVERAGE	CRU PR	BLOS
TRIAL 1	TGT 1	1 HITS	14 REPAIRS	0			
TRIAL 2	TGT 1	1 HITS	1 REPAIRS	3			
TRIAL 3	TGT 1	1 HITS	25 REPAIRS	3			
TRIAL 4	TGT 1	1 HITS	12 REPAIRS	12			
TRIAL 5	TGT 1	1 HITS	15 REPAIRS	0			
TRIAL 6	TGT 1	1 HITS	4 REPAIRS	4			
TRIAL 7	TGT 1	1 HITS	15 REPAIRS	0			
TRIAL 8	TGT 1	1 HITS	12 REPAIRS	0			
TRIAL 9	TGT 1	1 HITS	11 REPAIRS	11			
TRIAL 10	TGT 1	1 HITS	18 REPAIRS	3			
TRIAL 11	TGT 1	1 HITS	22 REPAIRS	3			
TRIAL 12	TGT 1	1 HITS	27 REPAIRS	2			
TRIAL 13	TGT 1	1 HITS	1 REPAIRS	0			
TRIAL 14	TGT 1	1 HITS	19 REPAIRS	0			
TRIAL 15	TGT 1	1 HITS	5 REPAIRS	14			
TRIAL 16	TGT 1	1 HITS	9 REPAIRS	0			
TRIAL 17	TGT 1	1 HITS	3 REPAIRS	3			
TRIAL 18	TGT 1	1 HITS	1 REPAIRS	1			
TRIAL 19	TGT 1	1 HITS	21 REPAIRS	3			
TRIAL 20	TGT 1	1 HITS	2 REPAIRS	0			
TRIAL 21	TGT 1	1 HITS	14 REPAIRS	1			
TRIAL 22	TGT 1	1 HITS	16 REPAIRS	16			
TRIAL 23	TGT 1	1 HITS	21 REPAIRS	1			
TRIAL 24	TGT 1	1 HITS	1 REPAIRS	1			
TRIAL 25	TGT 1	1 HITS	1 REPAIRS	1			

The 'total number of hits' and the minimum number of craters requiring repair to obtain a 4000' x 50' operating surface, are listed for each type #1 target that receives at least one hit

This listing is useful for plotting the distribution of the trial results when the average values and standard deviation, given below, are insufficient.

TARGET TYPE	AVG. CRU SFD. DLY. COVERAGE	AVG. PCMB OTHER EMP. COVERAGE	CRU PR	BLOS
1	0.0	0.0	0.0	0.0
2	6.4	6.05	0.0	0.0
	20.0	0.0	0.0	0.0

**DAMAGE STATISTICS BY TARGET TYPE**

TARGET TYPE	AVERAGE PERCENT HIT	STANDARD DEVIATION	COVERAGE SFD	COVERAGE OTHER	COVERAGE CRU
1	82.0	24.5	0.0	0.0	0.0

AT LEAST ONE MINIMUM RUNWAY SECTION WAS OPEN AFTER 16.0 PERCENT OF THE ATTACKS

WHEN ALL RUNWAYS WERE CLOSED, 2.0% (1.0%) CRATERS REQUIRED REPAIR, ON THE AVERAGE, TO PROVIDE A MINIMUM RUNWAY

Standard deviation  
Note that the repair requirement is not averaged over all trials but only those in which the minimum surface was not available.

Fig. 15 — Trial-by-trial results and target damage statistics, Case 4

Total computation time for these four cases was just under 53 sec on USAF's WWMCCS Honeywell 6060 computer. If not overlaid, 36K words of core are required for AIDA as presently dimensioned, that is, for 250 targets, 50 attacks, 20 target types, and 10 weapon types.



Appendix A

DETAILED DESCRIPTION OF AIDA INPUT

The basic types of input cards employed with AIDA are as noted below:

CONT	control card
TGT	target card; one per target
ATT	attack card; one per weapon delivery pass (or group of identical passes)
ATT2	alternate attack card
EMD	effective miss distance card; one for each weapon type
REDO	controls sequential cases
END	terminates overall computation

The ATT2 card is actually two cards in sequence and the EMD card may have up to three supplementary cards. A detailed description of the entries for each type of card is presented on the pages that follow.

The general arrangement of data on all basic card types is similar; the card type-name is placed (left-adjusted) in the first four columns and the data are listed in eleven 6-column fields between Columns 7 and 72. All data are read with a F6.0 format; i.e., they are to be real numbers. If a whole number is to be input, it may be entered (right-adjusted) in the field without a decimal point; the decimal point is necessary otherwise. Columns 5 and 6 on the ATT, ATT2, and EMD cards are also used, as will be described, and the name of the target complex being studied and a name for each target may be included in Columns 73 through 80 of the CONT and TGT cards, respectively; any alphanumeric names are acceptable.

All linear dimensions should be in consistent units\* (e.g., feet) and the target orientation and the attack heading entries should be in degrees.

---

\* If ATT2 cards are to be used, all linear dimensions must be in feet.

CONT

The CONT card controls the mode of operation, the choice of random number generator, the number of trials (attack replications), and print-out options; specifies the minimum clear length (MCL) and minimum clear width (MCW) for runway attack effectiveness calculations; and controls the runway repair assessment.

<u>Columns</u>	<u>Data Entry</u>
1-4	CONT
11-12	When 0, the seed for the random number generator is the same for all runs. If greater than 0, the seed is changed from run to run; if equal to -1, the random number generator is locked out. If equal to -2, the expected-value mode of operation replaces the Monte Carlo mode.
13-18	Desired number of replications. Default is 1.
23-24	Controls printout options as follows. If entry is 5 Prints multiple trial statistics plus a condensed listing of hits by trial 4 Prints multiple trial statistics plus a condensed listing of runway status by trial 3 Prints multiple trial statistics only 2 Above plus runway results for each trial 1 All above plus hit summary for each trial 0 All above plus stored hit data for each trial -1 All above plus all hits and target corners -2 All above plus all impact points
30	Controls printout of intermediate information for program test purposes; should normally be 0. If set to greater than 7, the random number generator is locked out. See the program source listing for the effect of other values.
31-36	MCL for aircraft operations. (Used to test if the runways are open.)
37-42	MCW for aircraft operations. (Used to test if the runways are open.)

<u>Columns</u>	<u>Data Entry</u>
48	When entry is 1, runway results will include the minimum number of craters to be repaired for the runway to meet the MCL and MCW criteria.
54	When the entry is 1, a plot of all impact points will be included for all closed runways (if, also, the printout option entry in Columns 23 and 24 is less than 3); when the entry is 2, impact plots are provided for each runway whether or not it is closed.
55-60	The distance that the "minimum runway rectangle" is to be shifted laterally in checking for an adequate section; the default value is 5.
61-66	The distance along the runway that the minimum runway rectangle is to be shifted in checking for an adequate section; the default value is 250.
73-80	A name can be entered here for the entire target complex and it will appear in the heading of the output listing.

TGT

Each TGT card designates the location, size, and orientation of a rectangular target.

<u>Columns</u>	<u>Data Entry</u>
1-3	TGT
7-12	The X-coordinate of the westernmost corner of the target.
13-18	The Y-coordinate of the westernmost corner of the target. If a target boundary runs exactly north-south, the X and Y coordinates of the southwestern corner should be specified.
19-24	Target dimension along the boundary running northeast (or north) from the X and Y coordinates of the reference corner specified in the two previous fields.
25-30	Target dimension along the boundary running southeast (or east) from the reference corner.
31-36	Heading in degrees of the northeast (or north) heading boundary of the target (along the dimension specified in Columns 19 to 24). (Meaning varies for target type #21; see below.)
41-42	Target type. Targets may be grouped into up to 10 (or 20) different categories with like vulnerabilities. This entry is used in conjunction with the effective miss distance on the EMD card. Target type #1 is restricted to runways and taxiways that may be used for flight operations; there will be no more than 5 targets of this type. Entering a 21 for target type actually acts as a signal (but only in conjunction with the expected-value mode) directing that a $17 \times 17$ grid of hit-density values be tabulated over a square, the southwest corner of which is entered in Columns 7 to 12 and 13 to 18. In this case, entries in the third, fourth, and seventh fields have no meaning. Unless a different value is entered in Columns 31 to 36 (preferably a number divisible by 16), the default dimension of the square is 4000, for a grid increment of 250.

Columns

Data Entry

There may be one or more target type #21 cards, and they may be intermingled with normal target cards; however, when present, one of the type #21 cards *must* be the last target card entered for a case.

48

If greater than 0, all hit locations will be saved (and printed when entry in Column 24 of the CONT card is 0 or less).

73-80

A name or number for the target (any alphanumeric) may be entered here. This name as well as the sequence number that is assigned automatically will appear for target identification in the output listing.

ATT

The ATT card specifies the parameters of each weapon-delivery pass. Inputs required are the attack heading (measured from north in the coordinate system used to specify the targets), the desired mean point of impact (DMPI) for a single weapon or for the middle of a stick of weapons, the aiming error expressed as REP and DEP, the ballistic error of the individual weapons, the number of weapons to be delivered in the pass, the stick length, and the weapon type (related to the effective miss distance on the EMD card).

<u>Columns</u>	<u>Data Entry</u>
1-3	ATT
5-6	Total number of passes with the following characteristics; default = 1.
10-12	Attack heading in degrees from north.
13-18	The X-coordinate of the DMPI of a single weapon or the middle of a stick of weapons.
19-24	The Y-coordinate of the DMPI as above.
25-30	The REP.
31-36	The DEP.
37-42	Ballistic dispersion in range of individual weapons (R-DISP).
43-48	Ballistic dispersion in deflection of individual weapons (D-DISP). Default value is R-DISP.
49-54	The number of weapons in the stick.
55-60	The length of the stick (the distance between the first and last weapon of the stick in the absence of dispersion).
61-66	The weapon type (used in effectiveness calculations together with EMD and target type). An entry is required (an integer from 1 to 10); otherwise hits will not be recorded.
67-72	Probability of arrival at target; default = 1.0.

ATT2

The ATT2 card should be used in place of the ATT card when the user wishes assistance with trajectory calculations. When this card is used the user expresses the attack in terms of speed, altitude, dive angle, intervalometer settings, etc., and a special subroutine converts these inputs to those demanded on the ATT card. The conversion procedure is the JMEM/AS Open End Method Zero as outlined in the *Users Manual for JMEM/AS Open-End Methods*, WANG Labs., Inc., Tewksbury, Mass., August 1974.

Both ATT and ATT2 type cards may be used in the same run; the order of entry is of no importance. When ATT2 cards are used the input data will be reproduced as submitted, as well as being tabulated in the normal manner, after conversion.

Data input with the ATT2 procedure require *two* cards. The first card is labeled ATT2 in the first 4 columns and has input similar to that on an ATT card (all fields are read with a F6.0 format); a second unlabeled card is mandatory following each ATT2 card. The format for both cards follows. When these cards are used, all linear dimensions in the input data will be in feet.

<u>Columns</u>	<u>Data Entry</u>
1-4	ATT2
5-6	Total number of passes with the following characteristics; default = 1.
10-12	Attack heading in degrees from north.
13-18	The X-coordinate of the DMPI of a single weapon or the middle of a stick of weapons.
19-24	The Y-coordinate of the DMPI as above.
25-30	The CEP in the normal plane in <i>mils</i> , or, if DEP is specified, a constant which, when divided by the sin of the impact angle, gives the REP, in <i>mils</i> .
31-36	The DEP in <i>mils</i> (if omitted, CEP controls).
37-42	Ballistic dispersion in <i>mils</i> .
49-54	The number of weapons in the stick.
61-66	The weapon type.
67-72	Probability of arrival at target; default = 1.0.

The data format for the second card of each ATT2 pair is as noted below (this card is used with a 6F6.0, 3F6.3 format). Typical ballistic data required for this card are noted in Table A-1.

<u>Columns</u>	<u>Data Entry</u>
7-12	Aircraft velocity (kn).
13-18	Release altitude of last bomb (ft).
19-24	Dive angle at release (deg).
25-30	Terminal velocity of weapon (cluster) or first leg of a high-drag bomb (ft/sec) (VT1 in JMEM).*
31-36	Terminal velocity of a cluster bomblet or a high-drag bomb (ft/sec) (VT2 in JMEM).
37-42	Probable error in estimating and correcting for wind effects (ft/sec).
43-48	Cluster opening time or fin opening time for a high-drag bomb (ms), or cluster/fin opening altitude (ft). (A <i>decimal</i> point is mandatory when altitude is input.) (TD or $H_f$ in JMEM.)
49-54	Intervalometer setting (ms).
55-60	Dispenser intervalometer setting (ms) (0 for clusters).

---

\* Illustrative values are noted on Table A-1.



Table A-1

TYPICAL BALLISTIC PARAMETERS

Weapon	VT <sub>1</sub> (fps)	Ballistic Parameter	
		VT <sub>2</sub> (fps)	T <sub>D</sub> or H <sub>f</sub>
Mk-81 Mod 1	1850	0	0
Mk-81 SE	1100	208	300 ms
Mk-82 Mod 1	1900	0	0
Mk-82 SE	1200	240	350 ms
Mk-83	2500	0	0
Mk-84	2850	0	0
M-117 Unretarded	1950	0	0
M-117 Retarded	900	168	300 ms
M-118	2450	0	0
AN-M64A1	1600	0	0
AN-M65A1	2000	0	0
Mk-36 DST	1200	240	350 ms
CBU-38	450	0	0
CBU-52B/B	1000	230	Variable Altitude (ft)
CBU-58/B	950	215	Variable Altitude (ft)

SOURCE: *Users Manual for JMEM/AS Open-End Methods*,  
WANG Labs., Inc., Tewksbury, Mass., August 1974.

EMD

The EMD card is optional and provides information regarding weapon performance against the various types of targets. The entries for this card are different for point-impact weapons and for CBU-type munitions. For point-impact weapons, a hit is assessed for any impact within a distance of EMD from the target. For CBU munitions, the EMD card is used to specify the dimensions of the rectangular bomblet pattern.

The methods for expressing weapon coverage also differ for the two types of munitions. For point-impact weapons the EMD is also used as the weapon kill radius, and coverage is determined as that fraction of the target area that is covered by a circle of that radius.

For point-impact weapons (GP bombs or PGMs) the entries are:

<u>Columns</u>	<u>Data Entry</u>
1-3	EMD
5	Enter 1 if data are to be entered for 20 target types.*
6	Enter 1 if data on weapon reliability, $p_k$ , or effective kill radius for this weapon type, are to be entered (on the following card).
11-12	Weapon type (used in conjunction with Columns 61 to 66 on ATT card).
13-18	EMD for point-impact weapons versus target type #1.
19-24	EMD versus target type #2.
67-72	EMD versus target type #10.

If the weapons are CBU-type munitions, use the following entries on the EMD card.

<u>Columns</u>	<u>Data Entry</u>
1-3	EMD
5	Enter 1 if data are to be entered for 20 target types.

---

\* When more than 10 target types are involved, the EMD data and, if specified, the supplemental coverage data (see p. 48), for target types #11 through #20 are entered in ten 6-column fields from Columns 13 to 72 on cards that immediately follow the EMD card (and supplementary card).

<u>Columns</u>	<u>Data Entry</u>
6	Enter 1 if data on weapon reliability and/or on kill probabilities are to be entered for any target type on the following card.
11-12	Weapon type (used in conjunction with Columns 61 to 66 on ATT card).
13-18	Enter CBU pattern length as a <i>negative</i> entry.
19-24	Enter CBU pattern width as a <i>positive</i> entry.

SUPPLEMENTAL CARD FOR WEAPON RELIABILITY AND COVERAGE FACTORS

If a 1 is entered in Column 6 of an EMD card, a supplemental card must be included next with the weapon reliability and a set of entries for the several target types. Note that this card is not identified, but one must follow each EMD card that has an entry in Column 6. If a 1 is entered in Column 5 of an EMD card, as well as in Column 6, a second supplementary card is required for target types #11 through #20; this card is the fourth of four.

All entries on these cards are optional; the default value for reliability is 1.0. If an entry is made in any of the last 10 (20) fields and it is not greater than unity, it is taken as the user estimate of the  $p_k$  for that particular weapon-target combination for either point-impact weapons or CBU munitions. For point-impact weapons, an entry that exceeds unity is taken as an additional kill radius and another coverage fraction is determined as that fraction of the target area that is covered by a circle of that radius, given a hit within EMD of the target. Thus, when there are entries on the supplemental card for certain target types, coverage fractions are computed both for the corresponding value of EMD as well as for the value on the supplemental card.

<u>Columns</u>	<u>Data Entry</u>
7-12	Reliability* of this weapon type; default = 1.0.
13-18	$p_k$ or kill radius† for this weapon versus target type #1.
19-24	$p_k$ or kill radius for this weapon versus target type #2.
	.
	.
	.
67-72	$p_k$ of this type weapon versus target type #10.

Entries for target types #11 through #20 on a second supplemental card will be in the ten 6-column fields between Column 13 and Column 72.

\* Since these entries are read with an F6.0 format, the decimal point must be included.

† Only for point-impact weapons.

REDO

The REDO card is used to terminate the input for one case and initiate a new case with some or all of the previous inputs, as described earlier.

<u>Columns</u>	<u>Data Entry</u>
1-4	REDO
7-12	Number of prior targets to be retained. All will be retained if there is no entry. Use a negative entry if none are to be retained.
13-18	Number of prior attacks to be retained. All will be retained if there is no entry. Use a negative entry if none are to be retained.
19-24	An entry of unity suppresses the input listings for targets and/or for attacks and weapons if no changes have been made in these data sets from the prior case.

END

An END card must be included at the end of all data entry cards.

<u>Columns</u>	<u>Data Entry</u>
----------------	-------------------

1-3	END
-----	-----

Appendix B

GLOSSARY OF TERMS USED IN AIDA

KEY VARIABLES

INL	Distance along the runway the "minimum runway rectangle" is shifted.
INW	Lateral distance the minimum runway rectangle is shifted in checking for an adequate section.
ITRIAL	Number of the current trial.
KCBU	Switch; set to unity if any weapons are CBU's.
KPTI	Switch; set to unity if any weapons are the point-impact type.
KTEST	Index controlling variety of debugging printout options.
LIST	Switch; when set to unity, target and/or attack input lists are suppressed when unchanged.
MCL	Minimum adequate length for required runway.
MCR	Switch; set to unity when runway availability is to be checked.
MCW	Minimum adequate width for required runway.
MODE	Index controlling mode of operation (see pp. 9 and 38).
MTT	Largest target type number in the target array.
NA	Total number of weapon-delivery passes.
NAM	Maximum permissible number of weapon-delivery passes.
ND	Number of types of weapons in overall attack.
NHITD	Switch; set to unity when the expected-value mode is specified.
NJMEM	Number of weapon-delivery passes that required trajectory calculations.

NPLOT        Switch; set to 1 or 2 if runway impact plots are desired.

NPRINT       Index controlling results output (see pp. 9 and 38).

NREDO        Switch; set to unity if an additional case is specified.

NREP         Switch; set to unity when repair requirements are to be assessed.

NSAVE1       Number of targets to be retained for a subsequent case.

NSAVE2       Number of weapon-delivery passes to be retained for a subsequent case.

NST          Maximum number of targets for which hits can be stored.

NSTAT        Cumulative number of trials in which the minimum runway was available.

NT          Total number of targets.

NTM         Maximum permissible number of targets.

NTRIAL       Total number of trials specified.



KEY ARRAYS

AMD(I,J,K)      Weapon effectiveness data.

    I            Weapon type.

    J            Target type.

    K = 1        Effective miss distance.

          2        Effective damage radius or probability of kill.

ATT(I,J)        Strong array for weapon-delivery data.

    I            Weapon-delivery pass number; numbered internally in  
                  order of entry.

    J = 1        Heading (deg).

          2        X coordinate of desired mean point of impact.

          3        Y coordinate of DMPI.

          4        Range error probable of DMPI.

          5        Deflection error probable of DMPI.

          6        Dispersion in range (ground plane).

          7        Number of weapons released in pass.

          8        Length of stick (in ground plane).

          9        Weapon type.

         10        Dispersion in deflection.

         11        Probability attacker arrives at target.

CBUHT(J,K)      Impact coordinates of the centroid of the *J*th CBU  
                  pattern.

    K = 1        X coordinate.

          2        Y coordinate.

COV(L)          Fraction of target L covered by one or more CBU patterns.

HIT(I,J,K) Storage array for hit locations on specified targets.

I *I*th of those targets for which hit data are to be stored.

J = 1 X coordinate.

2 Y coordinate.

3 Weapon type.

K Number of hit on the *I*th target.

HITR(I,J,K) Storage array for hit locations on type #1 targets (i.e., runways and taxiways).

I,J,K See HIT(I,J,K).

IR(N) Switch; set to unity if the *N*th weapon-delivery attacker fails to reach target.

IZONE(K,J) Denotes which of the ordered targets (see T0) fall in the *K*th target zone.

J = 1 Lowest numbered target in the *K*th zone.

2 Highest numbered target in the *K*th zone.

MHRT(K) Target number of the *K*th target for which hit location data are to be stored.

MSTAT(J) Storage array for accumulating trial results of runway availability tests.

J = 1 Minimum number of repairs required to open a minimum runway.

2 Square of J = 1, above.

3-8 Not used.

MTYPE(I) Index that specifies whether or not supplementary data are to follow the EMD card for weapon type I.

NCBU(L) Number of CBU weapon patterns that cover all or part of target L.

NHIT(L)            Number of hits on target L; by both point-impact and CBU weapons.

NRW(I)            Target number of the *I*th runway entered.

P(L,K)

    K = 1            Expected fraction of target L that is covered by the effects of point-impact weapons; or probability of kill of target L due to point-impact weapons.

          2            Probability of kill of target L due to CBU weapon patterns.

STAT(L,J)         Storage array for accumulating trial results.

    L                Target number.

    J = 1            Number of hits by point-impact weapons.

          2            Square of J = 1, above.

          3            Trials with at least one hit.

          4            Fractional coverage by CBU weapons.

          5            Square of J = 4, above.

          6            Expected fractional coverage by point-impact weapon; or probability of kill.

          7            Probability of kill by CBU-type weapons.

          8            Unused.

STAT2(I,J)        Storage array for accumulating trial results for targets of a given type.

    I                Target type.

    J = 1            Fraction of the targets of type I that received at least one hit.

          2            Square of J = 1, above.

TGT(L,J)          Storage array for target data.

L Target number; numbered internally in order of entry.

J = 1 X coordinate of westernmost corner (#1).

2 Y coordinate of corner #1.

3 X coordinate of corner #2.

4 Y coordinate of corner #2.

5 X coordinate of corner #3.

6 Y coordinate of corner #3.

7 X coordinate of corner #4.

8 Y coordinate of corner #4.

9 Heading of northeast target leg.

10 Target type.

11 Switch; hits stored when reset to unity.

12 Dimension of NE target leg.

13 Dimension of SE target leg.

TO(I,J) Target order array in which targets are ordered according to increasing values of the sum of the coordinates of the western corner.

I Ith target in the ordered array.

J = 1 Value of (X+Y) for the Ith ordered target.

2 Number of the target as initially entered.

WPNREL(I) Reliability of weapon type I.

Appendix C

PROGRAM LISTING

```
1. C MAIN-AIDA AIRBASE DAMAGE ASSESSMENT MODEL
2. COMMON /ARRAYS/ TGT(250,13), ATT(50,11),AMD(10,20,2), TO(250,2),
3. XIZONE(50,2),NHIT(250),MHIT(20),MIT(20,3,25),NAW(5),HITP(5,3,250)
4. X , P(250,3),COV(250),MTYPE(10),NAME(250,2),WPNREL(10),NCBU(250)
5. COMMON /INT/NT,NA,ND,NM,KTEST,MCA,MCA,MCL,MODE,NPRINT,NAM,NST,MTT
6. COMMON /STATS/INITIAL,ITRIAL,NSTAT,STAT(250,8),STAT2(20,5),MSTAT(8)
7. COMMON /CNTL/ NREP,NPLOT,INW,INL,NSAVE1,NSAVE2,LIST,NJMEM
8. COMMON /HITN/ NRWD, NRWD
9. COMMON /CBUHT/ CBUHT(200,2), IR(50), KCRU, KPTI
10. NT = 0
11. C ***** NTM IS THE MAXIMUM NUMBER OF TARGETS *****
12. C ARRAYS: TGT, TO, NHIT, P, COV, NAME, NCBU, STAT
13. NTM = 250
14. C ***** NAM IS THE MAXIMUM NUMBER OF ATTACKS
15. C ARRAYS: ATT, IN
16. NAM = 50
17. C ***** NST IS THE NUMBER OF TARGETS FOR WHICH HITS CAN BE STORED.
18. C ARRAYS: HIT, MHIT
19. NST = 20
20. C ***** TO CHANGE ANY OF THE PRECEDING DIMENSIONS, MAKE THE APPROPRIATE
21. C ***** CHANGES IN THE ARRAYS AND THEN CHANGE THE LIMITING VALUE.
22. NA = 0
23. ND = 0
24. MCA = 0
25. INITIAL = 1
26. KTEST = 0
27. NRWD = 0
28. DO 5 I = 1,10
29. MTYPE(I) = 0
30. WPNREL(I) = 1.0
31. DO 5 J = 1,20
32. DC 5 K = 1,2
33. 5 AMD(I,J,K) = 0.0
34. 10 ITRIAL = 0
35. DO 12 I = 1, 5
36. 12 NRW(I) = 0
37. CALL INPUT
38. IF (.NHITC .EQ. 1) GO TO 40
39. IF (INITIAL .LT. 2) GO TO 25
40. NSTAT = 0
41. DO 15 I = 1,NT
42. DO 15 N = 1,5
43. 15 STAT(I,N) = 0.0
44. DO 20 M = 1,20
45. DO 20 A = 1,5
46. 20 STAT2(M,N) = 0.0
47. 25 DO 30 I = 1, NST
48. 30 MHIT(I) = 0
49. 40 CONTINUE
50. DO 45 N = 1,8
51. 45 MSTAT(N) = 0
52. CALL TSTDIM
53. IF (.NHITC .EQ. 0) GO TO 60
54. IF (TGT(NT,13) .EQ. 21.) CALL GRIDEN
55. DO 50 I = 1,NT
56. COV(I) = 0.0
57. P(I,1) = 0.0
58. P(I,2) = 0.0
59. 50 P(I,3) = 0.0
60. CALL EXPHIT
```

```
61.      GO TO 200
62.      60  IF (KPT1 .EQ. 0) GO TO 100
63.      CALL TGTCPN
64.      CALL TGTCON
65.      100  CONTINUE
66.      ITRIAL = ITRIAL + 1
67.      DO 105 I1 = 1, NT
68.      DO 105 I2 = 1, 3
69.      DO 105 I3 = 1, 25
70.      105  HIT(I1,I2,I3) = 0.0
71.      DO 110 I = 1, NT
72.      COV(I) = 0.0
73.      P(I,1) = 0.0
74.      P(I,2) = 0.0
75.      P(I,3) = 0.0
76.      NCBU(I) = 0
77.      110  NHIT(I) = 0
78.      DO 115 I1 = 1, 5
79.      DO 115 I2 = 1, 3
80.      DO 115 I3 = 1, 250
81.      115  HITR(I1,I2,I3) = 0.0
82.      CALL BOMB
83.      IF (KCRU .EQ. 1) CALL CRU
84.      IF (MCS .EQ. 0) GO TO 130
85.      CALL CHECKC
86.      130  CONTINUE
87.      IF (NTRIAL .LT. 2) GO TO 170
88.      DO 140 I = 1, NT
89.      AID = NHIT(I) - NCBU(I)
90.      STAT(I,1) = STAT(I,1) + AID
91.      STAT(I,2) = STAT(I,2) + AID*AID
92.      IF (AID .GT. 0.0) STAT(I,3) = STAT(I,3) + 1.
93.      STAT(I,4) = STAT(I,4) + COV(I)
94.      STAT(I,5) = STAT(I,5) + COV(I)*COV(I)
95.      STAT(I,6) = STAT(I,6) + P(I,1)
96.      STAT(I,7) = STAT(I,7) + P(I,2)
97.      STAT(I,8) = STAT(I,8) + P(I,3)
98.      140  CONTINUE
99.      DO 150 M = 1, 20
100.     MM = 0
101.     NN = 0
102.     AID1 = 0.0
103.     AID3 = 0.0
104.     AID4 = 0.0
105.     AID5 = 0.0
106.     DO 150 I = 1, NT
107.     IF (TGT(I,10) .NE. 0) GO TO 150
108.     NN = NN + 1
109.     NAID = NHIT(I) - NCBU(I)
110.     AREA = TGT(I,12) - TGT(I,13)
111.     AID1 = AID1 + AREA
112.     AID3 = AID3 + AREA*P(I,1)
113.     AID4 = AID4 + AREA*P(I,2)
114.     AID5 = AID5 + AREA*P(I,3)
115.     IF (NAID .GT. 0) MM = MM + 1
116.     150  CONTINUE
117.     IF (MM .EQ. 0) GO TO 160
118.     AID = MM/NAID
119.     STA = MM/AID
120.     STAT2(M,1) = STAT2(M,1) + STA
121.     STAT2(M,2) = STAT2(M,2) + STA*STA
```

```

122.          STAT2(M,3) = STAT2(M,3) + AID3/AID1
123.          STAT2(M,4) = STAT2(M,4) + AID4/AID1
124.          STAT2(M,5) = STAT2(M,5) + AID5/AID1
125.          160 CONTINUE
126.          170 CONTINUE
127.          IF (NPRINT .GT. 1) GO TO 180
128.          CALL PRINT
129.          180 CONTINUE
130.          IF (NPRINT .EQ. 5) GO TO 190
131.          DO 195 L = 1,NT
132.          185 WRITE(6,1001) ITRIAL, L, NHIT(L)
133.          190 CONTINUE
134.          IF (ITRIAL .LT. NTRIAL) GO TO 100
135.          IF (ITRIAL .GT. 1) CALL STATIS
136.          200 IF (NREDO .EQ. 1) GO TO 10
137.          STOP
138.          1001 FORMAT(' ',ITRIAL',14,' TGT',14,' HITS',14)
139.          END
140.          SUBROUTINE INPUT
141.          INTEGER *4 LABEL, AN, NAME1, NAME2, NAME, NBASE1, HBASE2
142.          COMMON /ARRAYS/ TGT(250,13), ATT(50,11),AMD(10,20,2), TO(250,2),
143.          XIZONE(50,2),NHIT(250),MHIT(20),HIT(20,3,25),NRW(5),HITR(5,3,250)
144.          X ,P(250,3),COV(250),MTYPE(10),NAME(250,2),WPNREL(10),NCBU(250)
145.          COMMON/INT/NT,NA,ND,NTM,KTEST,PCR,MCW,MCL,MJDL,NPRINT,NAM,NST,MTT
146.          COMMON/STATS/NTRIAL,ITRIAL,NSTAT,STAT(250,8),STAT2(20,5),MSTAT(8)
147.          COMMON /CONTROL/ NREP,NPLCT,INW,INL,NSAVE1,NSAVE2,LIST,NJMEM
148.          COMMON /CBUHIT/ CRUHT(200,2), IR(50), KCBU, KPTI
149.          COMMON / HITON / NHITD, NREDO
150.          DIMENSION LABEL(6), DATA(11)
151.          DATA LABEL /'TGT ','ATT ','ATT2','END ','CONT','REDO'/
152.          NJMEM = 0
153.          LIST1 = 0
154.          LIST2 = 0
155.          IF (NREDO .EQ. 0) NHITD = 0
156.          IF (NREDO .EQ. 0) GO TO 2
157.          C SPE NOTE AT LABEL '45'
158.          IF (NSAVE1 .GT. 0) NT = NSAVE1
159.          IF (NSAVE2 .GT. 0) NA = NSAVE2
160.          IF (NSAVE1 .LT. 0) NT = 0
161.          IF (NSAVE2 .LT. 0) NA = 0
162.          IF (NSAVE1 .EQ. 0) LIST1 = LIST
163.          IF (NSAVE2 .EQ. 0) LIST2 = LIST
164.          NSAVE1 = 0
165.          NSAVE2 = 0
166.          2 CONTINUE
167.          NREDO = 0
168.          6 READ (5,101) AN, NTYPE, (DATA(I), I=1,11), NAME1,NAME2
169.          C NO ENTRY IS REQUIRED IN COLUMNS 5 AND 6. IF AN INTEGER IS FOUND
170.          C ON AN ATTACK CARD, THE ATTACK WILL BE REPEATED SO THAT THERE WILL
171.          C BE THAT TOTAL NUMBER OF ATTACKS WITH THE STATED CHARACTERISTICS.
172.          C (ONE ATTACK IS ASSUMED IF THERE IS NO ENTRY.)
173.          C IF AN ENTRY IS NOTED ON AN AMD CARD, RELIABILITY AND/OR PROBABILITY
174.          C OF KILL DATA WILL BE EXPECTED ON THE NEXT CARD.
175.          IF (AN .EQ. LABEL(1)) GO TO 10
176.          IF (AN .EQ. LABEL(2)) GO TO 20
177.          IF (AN .EQ. LABEL(3)) GO TO 26
178.          IF (AN .EQ. LABEL(4)) GO TO 30
179.          IF (AN .EQ. LABEL(5)) GO TO 40
180.          IF (AN .EQ. LABEL(6)) GO TO 45
181.          GO TO 50
182.          10 NT = NT + 1

```

```
183.          LIST1 = 0
184.          IF (NT .GT. NNM) GO TO 120
185.          DO 12 I = 1,2
186.    12      TGT(NT,I) = DATA(I)
187.          DO 13 I = 3,4
188.    13      TGT(NT,I+9) = DATA(I)
189.          DO 14 I = 5,7
190.    14      TGT(NT,I + 4) = DATA(I)
191.          NAME(NT,1) = NAME1
192.          NAME(NT,2) = NAME2
193.          GO TO 6
194.    20      NA = NA + 1
195.          LIST2 = 0
196.          IF (NA .GT. NNM) GO TO 130
197.          DO 22 I = 1,6
198.    22      ATT(NA,I) = DATA(I)
199.          ATT(NA,10) = DATA(7)
200.          IF (DATA(7) .EQ. 0.0) ATT(NA,10) = ATT(NA,6)
201.          IF (DATA(11) .EQ. 0.1) DATA(11) = 1.0
202.          ATT(NA,11) = DATA(11)
203.          DO 24 I = 7,9
204.    24      ATT(NA,I) = DATA(I+1)
205.          NTYPE = MAX0((NTYPE-1),0)
206.          IF (NTYPE .EQ. 0) GO TO 6
207.          GO TO 20
208.    26      NA = NA + 1
209.          LIST2 = 0
210.          IF (NA .GT. NNM) GO TO 130
211.          IF (DATA(11) .EQ. 0.0) DATA(11) = 1.0
212.          CALL JMEMO(INJMEM,DATA)
213.          DO 27 I = 1,11
214.    27      ATT(NA,I) = DATA(I)
215.          NTYPE = MAX0((NTYPE-1),0)
216.          IF (NTYPE .EQ. 0) GO TO 6
217.          NA = NA + 1
218.          DO 29 I = 1,11
219.    29      ATT(NA,I) = ATT(NA-1,I)
220.          GO TO 26
221.    30      ND = ND + 1
222.          LIST2 = 0
223.          N = DATA(11)
224.          IF (N .GT. 10) GO TO 140
225.          NTYPE(N) = NTYPE
226.          DO 31 N = 2,11
227.    31      AND(N,N-1,1) = DATA(N)
228.          IF ((NTYPE .EQ. 0) .OR. (NTYPE .EQ. 10)) GO TO 34
229.          READ(5,114) WR,(DATA(I), I=1,10)
230.          WPNPCL(N) = WR
231.          IF (WR .EQ. 0.0) WPNPCL(N) = 1.0
232.          DO 32 N = 1,10
233.    32      AND(N,N,2) = DATA(N)
234.          34 CONTINUE
235.    C THE FOLLOWING 8 STATEMENTS PERMIT 20 TARGET TYPES WHEN NEEDED
236.          IF (NTYPE .LT. 10) GO TO 6
237.          READ(5,114) (DATA(I), I = 1,11)
238.          DO 36 N = 11,20
239.    36      AND(N,N,1) = DATA(N-9)
240.          IF (NTYPE .LT. 11) GO TO 6
241.          READ(5,114) (DATA(I), I = 1,11)
242.          DO 38 N = 11,20
243.    38      AND(N,N,2) = DATA(N-9)
```



```
244.          GO TO 6
245.          40  MODE = DATA(1)
246.             IF (MODE.EQ.0) CALL RSTART(7)
247.             NTRIAL = DATA(2)
248.             NPRINT = DATA(3)
249.             KTEST = DATA(4)
250.             IF (NTRIAL.LT. 2)  NTRIAL = 1
251.             MCR = 0
252.             NHITO = 0
253.             IF (MODE.EQ. -2)  NHITO = 1
254.             IF (NHITO.EQ. 1)  DATA(5) = 0.0
255.             IF (DATA(5).GT. 0.0)  MCR = 1
256.             MCL = DATA(5)
257.             MCW = DATA(6)
258.             NREP = DATA(7)
259.             NPLDT = DATA(8)
260.             INW = DATA(9)
261.             INL = DATA(10)
262.             IF (INW.EQ. 0)  INW = 5
263.             IF (INL.EQ. 0)  INL = 250
264.             NBASE1 = NAME1
265.             NBASE2 = NAME2
266.             GO TO 6
267.          45  NREDO = 1
268.          C   THE FIRST TWO DATA ENTRIES ON THE REDO CARD MAY BE USED TO SPECIFY
269.          C   THE NUMBERS OF PRIOR TARGETS AND ATTACKS, RESPECTIVELY, TO BE
270.          C   INCLUDED IN THE NEW CALCULATION.  THOSE NUMBERS WILL BE SELECTED
271.          C   IN RANK ORDER FROM THOSE INPUT PREVIOUSLY; IN NO CASE MAY THE
272.          C   NUMBER BE LARGER THAN THE NUMBER AVAILABLE.  IF NO NUMBER IS
273.          C   ENTERED, ALL PRIOR TARGETS AND ATTACKS WILL BE INCLUDED.  IF A
274.          C   NEGATIVE NUMBER IS ENTERED (EG -1), NONE OF THE PRIOR MEMBERS
275.          C   OF THAT SET WILL BE TREATED.
276.             NSAVE1 = DATA(1)
277.             NSAVE2 = DATA(2)
278.          C   IF THE THIRD ENTRY (COL 24) IS SET TO UNITY, THE TARGET LIST
279.          C   AND/OR THE ATTACK/WEAPON LISTS WILL BE SUPPRESSED IF
280.          C   THEY HAVE NOT BEEN CHANGED.
281.             LIST = DATA(3)
282.          50  CONTINUE
283.             WRITE (6,111) NTRIAL, NPRINT, MODE, MCL, INL, MCW, INW,
284.          X   NREP, NPLDT, KTEST
285.             IF ((NBASE1.EQ. 0).AND. (NBASE2.EQ. 0))  GO TO 55
286.             WRITE (6,100)  NBASE1, NBASE2
287.          55  CONTINUE
288.             IF (LIST1.EQ. 1)  GO TO 65
289.             WRITE (6,102)
290.             MTT = 0
291.             DO 60 I = 1,NT
292.             IF ((TGT(I,10).GT. MTT).AND. (TGT(I,10).NE. 21.))
293.          X   MTT = TGT(I,10)
294.          60  WRITE (6,112) I,(TGT(I,J),J=1,2),(TGT(I,J),J=12,13),
295.          X   (TGT(I,J),J=9,11),NAME(I,1),NAME(I,2)
296.          65  IF (LIST2.EQ. 1)  GO TO 95
297.             WRITE (6,104)
298.             MWPV = 0
299.             KPTI = 0
300.             DO 70 I = 1, NA
301.             IF (ATT(I,9).GT. MWPV)  MWPV = ATT(I,9)
302.             IF (AMD(ATT(I,9),1,1).GE. 0)  KPTI = 1
303.          70  WRITE (6,103) I, (ATT(I,J),J=1,6),ATT(I,10),(ATT(I,J),J=7,9)
304.          X   , ATT(I,11)
```

```
305.      IF (ND .EQ. 0) GO TO 95
306.      WRITE (6,105) (I, I=1,10)
307.      DO 90 I = 1, MWPN
308.      WRITE(6,106) I, WPNREL(I),(AMD(I,J,1),J=1,10)
309.      IF ((MTYPE(I) .EQ. 0) .OR. (MTYPE(I) .EQ. 10)) GO TO 80
310.      WRITE (6,107) (AMD(I,J,2), J = 1,10)
311.      80 CONTINUE
312.      IF (MTYPE(I) .LT. 10) GO TO 90
313.      WRITE(6,115) (AMD(I,J,1), J=11,20)
314.      IF (MTYPE(I) .EQ. 10) GO TO 90
315.      WRITE(6,107) (AMD(I,J,2), J=11,20)
316.      90 CONTINUE
317.      95 CONTINUE
318.      C TARGET TYPE #1 IS RESERVED FOR RUNWAYS AND TAXIWAYS (OR OTHER
319.      C LARGE TARGETS IF MCP=0) AND HIT STORAGE IS PROVIDED FOR 250 HITS
320.      C BUT FOR A MAXIMUM OF FIVE TARGETS OF TYPE #1.
321.      NTX = 0
322.      DO 99 I = 1, NT
323.      IF (TGT(I,10) .NE. 1.) GO TO 99
324.      NTX = NTX + 1
325.      IF (NTX .GT. 5) GO TO 150
326.      NRW(NTX) = I
327.      99 CONTINUE
328.      RETURN
329.      120 WRITE (6,108)
330.      STOP
331.      130 WRITE (6,109)
332.      STOP
333.      140 WRITE (6,110)
334.      STOP
335.      150 WRITE (6,116)
336.      STOP
337.      100 FORMAT(' ',15X,'***** BASE COMPLEX NAME - ',2A4,' *****',/,/)
338.      101 FOPMAT( 4,12,11F6.0,2A4)
339.      102 FORMAT (' ', 20X,'TARGET DATA',/, ' NUMBER          X-DIM      Y-DIM
340.      X NE LIMB   SE LIMB      ANGLE TGT TYPE      STORE   BLOC NO',/,/)
341.      103 FORMAT(' ',14,2X,10F10.0,F10.3)
342.      104 FOPMAT(' ',/, 20X,'ATTACK DATA',/, ' NUMBER          HDG      X-DMPI
343.      X Y-DMPI    REP      CEP      R-DISP  D-DISP  NO WPNS  LENG
344.      XTH WPN TYPE  ARPIVAL',/,)
345.      105 FOPMAT('1', 35X,'MISS DISTANCES ALLOWED FOR EFFECTIVE HITS',/,
346.      X50X,'TARGET TYPES',/,20X,10(6X,12,2X),/, ' WPN TYPE  WPN REL ',
347.      X/,/,/)
348.      106 FORMAT(' ',16,7X,F5.3,10F10.0)
349.      107 FORMAT(' ', 21X,10F10.3)
350.      108 FORMAT(' ',/, '***** TOO MANY TARGETS HAVE BEEN SPECIFIED *****')
351.      109 FORMAT(' ',/, '***** TOO MANY ATTACKS HAVE BEEN SPECIFIED *****')
352.      110 FORMAT(' ',/, '***** TOO MANY TYPES OF WEAPONS HAVE BEEN ',
353.      X 'SPECIFIED *****')
354.      111 FORMAT ('1', 130('**'),/,
355.      X 40X,'*****AIDA***** AN AIRBASE DAMAGE ASSESSMENT MODEL',/,
356.      X 40X,'DEVELOPED BY THE RAMSTEIN OFFICE OF THE RAND CORPORATION',/
357.      X ,130('**'),/, ' NO OF TRIALS ', 13, ' NPRINT ',12, ' MODE ',12,
358.      X ' MCL ',16,(' ',14,') MCW ',14,(' ', 12,') MIN REPAIR ',
359.      X 11, ' PLOT HITS ',11, ' TEST ',12,/,130('**'),/,/)
360.      112 FORMAT (' ', 14,4X,7F10.0,2X,2A4)
361.      114 FORMAT ( 6X,11F6.0)
362.      115 FOPMAT (' ',18X, 10F10.0)
363.      116 FOPMAT (' ',/, '***** TOO MANY RUNWAYS/TAXIWAYS HAVE BEEN',
364.      X ' SPECIFIED ***-')
365.      END
```

```
366.          SUBROUTINE TGTDIM
367.          COMMON /ARRAYS/ TGT(250,13), ATT(50,11),AMD(10,20,2), TO(250,2),
368.          XIZONE(50,2),NHIT(250),MHIT(20),HIT(20,3,25),NRW(5),HITR(5,3,250)
369.          X ,P(250,3),COV(250),MTYPE(10),NAME(250,2),WPNREL(10),NCRU(250)
370.          COMMON/INT/NT,NA,ND,NTM,KTEST,MCR,MCW,MCL,MODL,NPRINT,NAM,NST,MTT
371.          COMMON/STATS/NTRIAL,ITPIAL,NSTAT,STAT(250,8),STAT2(20,5),MSTAT(8)
372.          COMMON / HITDN / NHITD, MREDO
373.          IF (NPRINT .LT. 0) WRITE (6,104)
374.          DO 20 I = 1, NT
375.          C   TGTDIM COMPUTES AND STORES THE LOCATION OF THE OTHER CORNERS.
376.             L1 = TGT(I,12)
377.             L2 = TGT(I,13)
378.             THETA = TGT (I, 9)/57.3
379.             S = SIN (THETA)
380.             C = CCS (THETA)
381.             L1S = L1*S
382.             L1C = L1*C
383.             L2S = L2*S
384.             L2C = L2*C
385.             TGT (I,3) = TGT (I, 1) + L1S
386.             TGT (I,4) = TGT (I, 2) + L1C
387.             TGT (I,5) = TGT (I, 3) + L2C
388.             TGT (I,6) = TGT (I, 4) - L2S
389.             TGT (I,7) = TGT (I, 5) - L1S
390.             TGT (I,8) = TGT (I, 6) - L1C
391.             IF ((KTEST .GT. 2) .OR. (NPRINT .LT. 0))
392.             X   WRITE (6,102) I,(TGT(I,K), K=1,8)
393.             20 CONTINUE
394.             IF (NHITD .EQ. 1) GO TO 50
395.             NR = 0
396.             DO 30 I = 1, NT
397.             C   FOR SPECIFIED TARGETS THE TARGET NUMBER IS STORED IN MHIT FOR
398.             C   LATER REFERENCE.
399.             IF ((TGT(I,11) .LT. 1.) .OR. (TGT(I,10) .EQ. 1.)) GO TO 30
400.             NR = NR + 1
401.             IF (NR .GT. NST) GO TO 80
402.             MHIT(NR) = I
403.             IF ((KTEST .GT. 4) .AND. (ITRIAL .LT. 2))
404.             X   WRITE (6,101) NR, MHIT(NR)
405.             30 CONTINUE
406.             IF (NR .EQ. NST) GO TO 50
407.             NR1 = NR+1
408.             DO 40 I = NR1, NST
409.             40 MHIT(I) = 0
410.             50 CONTINUE
411.             IF ((NPRINT .LT. 0) .OR. (KTEST .GT. 2)) WRITE(6,104)
412.             RETURN
413.             80 WRITE (6,103)
414.             STOP
415.             101 FORMAT( ' ', 'MHIT( ', I2, ') = ', I2)
416.             102 FORMAT(' ', 'TARGET CORNER : TGT # ', I4, '414X,F6.0,1X,F6.0))
417.             103 FORMAT('0', ' COMPUTATION STOPPED: HIT DATA SPACE ',
418.             X ' *REQUIRED FOR MORE THAN "NST" TARGETS')
419.             104' FORMAT ('1')
420.             END
421.          SUBROUTINE TGTORD
422.          COMMON /ARRAYS/ TGT(250,13), ATT(50,11),AMD(10,20,2), TO(250,2),
423.          XIZONE(50,2),NHIT(250),MHIT(20),HIT(20,3,25),NRW(5),HITR(5,3,250)
424.          X ,P(250,3),COV(250),MTYPE(10),NAME(250,2),WPNREL(10),NCRU(250)
425.          COMMON/INT/NT,NA,ND,NTM,KTEST,MCR,MCW,MCL,MODE,NPRINT,NAM,NST,MTT
426.          C   THIS ROUTINE CREATES AN ARRAY IN WHICH THE TARGET NUMBERS ARE
```

```
427. C ORDERED ACCORDING TO INCREASING VALUES OF (X+Y) OF THE INDEX CORNER
428. C OUTPUT VALUES : TQ(I,1) IS THE (X+Y) AND TQ(I,2) IS THE 'ORIGINAL'
429. C TARGET NUMBER; I.F. ITS POSITION IN THE INPUT LIST
430. DO 10 I=1,NT
431. C INITIALIZES TO (TARGET ORDER) ARRAY
432. IF (TGT(I,9) .GT. 45.) GO TO 5
433. TQ(I,1)=TGT(I,1)+TGT(I,2)
434. GO TO 8
435. 5 TQ(I,1)=TGT(I,7)+TGT(I,8)
436. 8 TQ(I,2)=I
437. 10 CONTINUE
438. DO 20 J=2,NT
439. NTEST=0
440. DO 15 K=2,NT
441. C REORGANIZES THE TQ ARRAY INTO INCREASING VALUES OF THE INDEX CORNER
442. I=NT-K+2
443. IF (TQ(I,1) .GE. TQ(I-1,1)) GO TO 15
444. NTEST=1
445. T=TQ(I-1,1)
446. TN=TQ(I-1,2)
447. TQ(I-1,1)=TQ(I,1)
448. TQ(I-1,2)=TQ(I,2)
449. TQ(I,1)=T
450. TQ(I,2)=TN
451. 15 CONTINUE
452. IF (NTEST .EQ. 0) GO TO 25
453. 20 CONTINUE
454. 25 CONTINUE
455. IF (KTEST .LT. 3) GO TO 40
456. DO 30 I = 1, NT
457. NTO = TQ(I,2)
458. 30 IF (KTEST .GT. 5) WRITE (6,101) I , NTO
459. 40 CONTINUE
460. NT1 = NT + 1
461. DO 50 I = NT1, NTM
462. DO 50 J = 1,2
463. 50 TQ(I,J) = 0.0
464. RETURN
465. 101 FORMAT (' ', ' RANK ',I3,' TARGET # ', I3)
466. END
467. SUBROUTINE TGTZON
468. COMMON /ARRAYS/ TGT(250,13), ATT(50,11),AMD(10,20,2), TQ(250,2),
469. XIZONE(50,2),NHIT(250),MHIT(20),HIT(20,3,25),NRW(5),HITR(5,3,250)
470. X ,P(250,3),COV(250),MTYPE(10),NAME(250,2),WPNREL(10),NCBU(250)
471. COMMON/INT/NT,NA,ND,NTM,KTEST,MCR,MCW,MCL,MODE,NPRINT,NAM,NST,MTT
472. DIMENSION IZ(250)
473. C TGTZON IDENTIFIES TARGET LOCATION IN TERMS OF ITS IZONE SO THAT THE
474. C SUBSEQUENT SEARCH PROCESS CAN BE REDUCED. CONSIDER THE ENTIRE
475. C TARGET AREA MAPPED BY LINES OF CONSTANT (X+Y). ALL TARGETS WITH '
476. C 'INDEX CORNER' (X+Y) FALLING INTO THE K TH 500 FOOT SEGMENT OF
477. C (X+Y) ARE IN THE K TH ZONE. THE ORDERED INDEX NUMBER FOR THE
478. C TARGET WITH THE LOWEST (X+Y) IN THE ZONE IS IZONE(K,1); THAT WITH
479. C THE HIGHEST, IS IZONE(K,2). IF THERE ARE NO TARGETS IN A ZONE, THE
480. C IZONE VALUES ARE BOTH EQUAL TO THE INDEX NUMBER OF THE LAST TARGET
481. C AS IZONE(K-1,2)).
482. DO 10 I=1,NT
483. AA = TQ(I,1)/500.
484. 10 IZ(I) = AA
485. IZONE(1,1) = 0
486. IZONE(1,2) = 0
487. IF (IZ(I) .NE. 0) GO TO 14
```

```
488.          K = 1
489.          IZONE(1,1) = 1
490.          IZONE(1,2) = 1
491.          DO 12 I = 2,NT
492.          IF (IZ(I) .GT. 0) GO TO 22
493.          12 IZONE(1,2) = I
494.          C ALL HITS IN ZONE #1 HERE.
495.          GO TO 30
496.          14 DO 16 K = 2,50
497.          IF (IZ(1) .EQ. (K-1)) GO TO 18
498.          IZONE(K,1) = 0
499.          16 IZONE(K,2) = 0
500.          18 CONTINUE
501.          IZONE(K,1) = 1
502.          IZONE(K-1,2) = 1
503.          IZONE(K,2) = 1
504.          C AT THIS POINT K IS ZONE OF FIRST HIT
505.          DO 20 I = 2,NT
506.          IF (IZ(I) .GT. (K-1)) GO TO 22
507.          20 IZONE(K,2) = I
508.          C ON TRANSFER TO '22' K IS FIRST OCCUPIED ZONE AND I IS FIRST HIT
509.          C IN (K+1) ZONE.
510.          22 CONTINUE
511.          N = I
512.          DO 28 I = N,NT
513.          C SKIP TO 26 IF HIT IN ZONE OF PRIOR HIT
514.          IF (IZ(I) .EQ. IZ(I-1)) GO TO 26
515.          24 K = K+1
516.          IZONE(K,1) = I-1
517.          IZONE(K,2) = I-1
518.          C IF NO HITS IN ZONE INCREMENT ZONE
519.          IF (IZ(I) .GT. (K-1)) GO TO 24
520.          IZONE(K,1) = I
521.          IZONE(K,2) = I
522.          GO TO 28
523.          C INCREMENT UPPER HIT IN ZONE
524.          26 IZONE(K,2) = I
525.          28 CONTINUE
526.          30 CONTINUE
527.          IF ((K+1) .GT. 50) GO TO 36
528.          C FILL ALL EXCESS ZONES
529.          K1 = K + 1
530.          DO 32 L = K1,50
531.          IZONE(L,1) = NT + 1
532.          32 IZONE(L,2) = NT + 1
533.          36 CONTINUE
534.          IF (KTEST .LT. 3) GO TO 50
535.          WRITE (6,101)
536.          DO 40 K = 1, 50
537.          40 WRITE (6,102) K, IZONE(K,1),IZONE(K,2),TO(IZONE(K,1),2),
538.          X TO(IZONE(K,2),2)
539.          50 CONTINUE
540.          RETURN
541.          101 FORMAT ('1', 'TARGETS BY ZONE',/,
542.          X'ZONE LOWER UPPER (LOWER UPPER)')
543.          102 FORMAT (' ', 14,4X,I3,4X,I3,7X,F4.0,3X,F4.0)
544.          END
545.          SUBROUTINE BOMB
546.          COMMON /ARRAYS/ TGT(250,13), ATT(50,11),AMD(10,20,2), TO(250,2),
547.          XIZONE(50,2),NHIT(250),MHIT(20),HIT(20,3,25),NRW(5),HITR(5,3,250)
548.          X ,P(250,3),COV(250),MTYPE(10),NAME(250,2),WPNREL(10),NCBU(250)
```

```
549. COMMON/INT/NT,NA,ND,NTM,KTEST,MCR,MC4,MCL,MODC,NPRINT,NAM,NST,MTT
550. COMMON /CBUHT/ CBUHT(200,2), IR(50), KCBU, KPTI
551. IF (KTEST .GT. 2) WRITE (6,102)
552. NCBUHT = 0
553. KCBU = 0
554. DO 40 I=1,NA
555. IR(I) = 0
556. NW = ATT(I,9)
557. FMD = AMD(NW,1,1)
558. IF ((MODE .LT. 0) .OR. (ATT(I,11) .EQ. 1.0)) GO TO 3
559. C IF (MODE .EQ. 0) GO TO 1
560. C RN = RANDT(1.)
561. C GO TO 2
562. 1 RN = RAN(1)
563. 2 IF (RN .LE. ATT(I,11)) GO TO 3
564. IR(I) = 1
565. GO TO 40
566. 3 CONTINUE
567. IF (EMD .LT. 0.0) KCBU = 1
568. DX=0
569. DY=0
570. NS=ATT(I,7)
571. PHI=ATT(I,1)/57.3
572. S=SIN(PHI)
573. C=COS(PHI)
574. SIGR=.483*ATT(I,4)
575. SIGD=1.483*ATT(I,5)
576. CALL GAUSS(SIGR,REFR)
577. CALL GAUSS(SIGD,DEFR)
578. AGZX=ATT(I,2)+RERR*S+DEFR*C
579. AGZY=ATT(I,3)+RERR*C-DEFR*S
580. BDGZX=AGZX-S*ATT(I,8)/2.
581. BDGZY=AGZY-C*ATT(I,8)/2.
582. IF (NS .LT. 2) GO TO 10
583. D=ATT(I,8)/(NS-1)
584. DX=S*D
585. DY=C*D
586. 10 CONTINUE
587. SIGX = ATT(I,6)
588. SIGY = ATT(I,10)
589. DO 20 M=1,NS
590. IF (EMD .LT. 0.0) NCBUHT = NCBUHT + 1
591. IF (NCBUHT .GT. 200) GO TO 60
592. IF (MODE .LT. 0) GO TO 13
593. C IF (MODE .EQ. 0) GO TO 11
594. C RN = RANDT(1.)
595. C GO TO 12
596. 11 RN = RAN(1)
597. 12 IF (RN .GT. WPNREL(NW)) GO TO 17
598. 13 CONTINUE
599. CALL GAUSS(SIGX,X)
600. CALL GAUSS(SIGY,Y)
601. BAGZX=BDGZX+X
602. BAGZY=BDGZY+Y
603. IF ((KTEST .GT. 2) .OR. (NPRINT .LT. -1))
604. X WRITE (6,101) I, M, BAGZX, BAGZY
605. IF (EMD .GE. 0.0) GO TO 16
606. CBUHT(NCBUHT,1) = BAGZX
607. CBUHT(NCBUHT,2) = BAGZY
608. GO TO 18
609. 16 IF ((BAGZX+BAGZY) .LT. -500.) GO TO 18
```

```
610.          INDEX = I
611.          CALL TESTHT(INDEX,BAGZX,BAGZY)
612.          GO TO 18
613.          17 IF (EMD .LT. J.0)   CBUHT(NCBUHT,1) = -10000.
614.          18 BDGZX=BDGZX+OX
615.          BDGZY=BDGZY+OY
616.          20 CONTINUE
617.          40 CONTINUE
618.          RETURN
619.          60 WRITE(6,103)
620.          STOP
621.          101 FORMAT(' ',/, 'ATTACK #',I4,' BOMB #',I4,' X-DIM ',F8.0,
622.          X ' Y-DIM ', F8.0)
623.          102 FORMAT (' ',30(' '), ' BOMB IMPACT AND HIT DATA ',30(' '),/,/)
624.          103 FORMAT('0', 'THE CBUHT ARRAY MUST BE ENLARGED TO ',
625.          X ' ACCOMMODATE MORE GBU WEAPONS')
626.          END
627.          SUBROUTINE TESTHT(I,BX,BY)
628.          COMMON /ARRAYS/ TGT(250,13), ATT(50,11),AMD(10,20,2), TC(250,2),
629.          X IZONE(50,2),NHIT(250),MHIT(20),HIT(20,3,25),NRW(5),HITR(5,3,250)
630.          X ,P(250,3),COV(250),MTYPE(10),NAME(250,7),WPNREL(10),NCRU(250)
631.          COMMON/INT/NT,NA,MD,NTM,KTEST,MCR,MCW,MCL,MJDE,NPRINT,NAM,NST,MTT
632.          XY = BX + BY
633.          NN=XY/500.
634.          K = MAX0(NN+1,1)
635.          K = MIN0(K,49)
636.          LL = 0
637.          IF (K .EQ. 1) GO TO 10
638.          LL = IZONE (K-1, 1)
639.          IF (K .EQ.2) GO TO 20
640.          IF ((IZONE(K-1,1) .EQ. IZONE(K-2,2)) .AND.
641.          X (IZONE(K-2,2) .NE. 1)) LL = IZONE(K,1)
642.          GO TO 20
643.          10 LL = IZONE (1, 1)
644.          20 CONTINUE
645.          LU = IZONE ((K+1), 2)
646.          IF (KTEST .GT. 3) WRITE (6,102) I, LL,LU
647.          DO 100 IL = LL, LU
648.          C CONSIDER ALL TARGETS BETWEEN THE LIMITS OF LL AND LU
649.          L = TC(IL,?)
650.          IF (L .LE. 0) GO TO 100
651.          IF ((TGT(L,12) + TGT(L,13)) .GT. 500.) GO TO 100
652.          D = 1.414*AMD(ATT(I,9),TGT(L,10),1)
653.          IF ((TGT(L,1) - D) .GT. BX) GO TO 100
654.          IF ((TGT(L,4) + D) .LT. BY) GO TO 100
655.          IF ((TGT(L,5) + D) .LT. BX) GO TO 100
656.          IF ((TGT(L,8) - D) .GT. BY) GO TO 100
657.          IF (KTEST .GT. 4) WRITE (6,101)I,L,BX,BY
658.          CALL HITGT(I,L,BX,BY)
659.          100 CONTINUE
660.          DO 120 L = 1, NT
661.          IF ((TGT(L,12) + TGT(L,13)) .LE. 500.) GO TO 120
662.          IF (KTEST .GT. 4) WRITE (6,101) I,L,BX,BY
663.          IL = L
664.          CALL HITGT(I,IL,BX,BY)
665.          120 CONTINUE
666.          130 CONTINUE
667.          RETURN
668.          101 FOFMAT (' ',10X,'TESTHT: ATTACK ',I3,' TGT ',I4,' X-DIM
669.          X ', F8.0, ' Y-DIM ', F8.0)
670.          102 FORMAT(' ',20X,'ATTACKER ',I3,' TARGET RANK LIMITS ',2I6,/)

```

```
671.          LND
672.          SUBROUTINE HITTGT(I,L,BX,BY)
673.          COMMON /ARPAYS/ TGT(250,3), ATT(50,11),AMD(10,20,2), TO(250,2),
674.          XIZONE(50,2),NHIT(250),MHIT(20),HIT(20,3,25),NRW(5),HITR(5,3,250)
675.          X ,P(250,3),COV(250),MTYPE(10),NAME(250,2),WPNREL(10),NCBU(250)
676.          COMMON/INT/NT,NA,ND,NTM,KTEST,MCR,MCW,MCL,MODE,NPRINT,NAM,NST,MTT
677.          DIMENSION DN(4),F(4),IS(4),D2(4)
678.          D = 0
679.          T = 0
680.          DD = AMD(ATT(I,9),TGT(L,10),1)
681.          S = SIN (TGT (L,9)/57.3)
682.          C = COS (TGT (L,9)/57.3)
683.          CO = C
684.          IF (DD .EQ. 0.0) GO TO 10
685.          T = (S + C)*DD
686.          D = (S - C)*DD
687.          10  X1 = TGT (L,1) - T
688.             Y1 = TGT (L,2) + D
689.             X2 = TGT (L,3) + D
690.             Y2 = TGT (L,4) + T
691.             X3 = TGT (L,5) + T
692.             Y3 = TGT (L,6) - D
693.             X4 = TGT (L,7) - D
694.             Y4 = TGT (L,8) - T
695.             IF ((BX .LT. X1) .OR. (BX .GT. Y3)) GO TO 100
696.             IF ((BY .LT. Y4) .OR. (BY .GT. Y2)) GO TO 100
697.             IF (TGT (L, 9) .EQ. 0.) GO TO 60
698.             T = S/C
699.             C = 1./T
700.             IF ((BX .LT. X2) .AND. (BY .GT. (Y1+C*(BX-X1)))) GO TO 100
701.             IF ((BX .GT. X2) .AND. (BY .GT. (Y2-T*(BX-X2)))) GO TO 100
702.             IF ((BX .GT. X4) .AND. (BY .LT. (Y4+C*(BX-X4)))) GO TO 100
703.             IF ((BX .LT. X4) .AND. (BY .LT. (Y1-T*(BX-X1)))) GO TO 100
704.          60  CONTINUE
705.             IF ((KTEST .GT. 2) .OR. (NPRINT .LT. 0)) WRITE(6,101)L,BX,BY
706.             NHIT(L) = NHIT(L) + 1
707.             NW = ATT(I,9)
708.             NTGT = TGT(L,10)
709.             IF ((NTGT .EQ. 1) .AND. (MCR .NE. 0)) GO TO 110
710.             PP = AMD(NW,NTGT,2)
711.          C  RESULTS INCLUDE AN ESTIMATE OF THAT FRACTION OF THE TARGET AREA THAT
712.          C  IS COVERED BY A CIRCLE OF RADIUS 'EMD'. IF A VALUE, NOT GREATER
713.          C  THAN ONE, IS INPUT WITH THE SUPPLEMENTARY EMD CARD, A HIT WILL BE
714.          C  ASSUMED TO ACHIEVE THAT FRACTIONAL KILL OF THE TARGET. IF A VALUE
715.          C  GREATER THAN ONE IS SPECIFIED, THAT VALUE WILL BE USED IN THE
716.          C  MANNER AS THE EMD TO COMPUTE AN ESTIMATE OF THE TARGET FRACTION
717.          C  THAT IS COVERED.
718.          C  COMPUTE DISTANCES NORMAL TO THE FOUR SIDES OF THE TARGET.
719.             IF (TGT(L,9) .EQ. 0.0) GO TO 75
720.             YD = Y1 + C*(BX-X1) - BY
721.             DN(1) = S*YD
722.             YD = Y2 - T*(BX-X2) - BY
723.             DN(2) = CO*YD
724.             YD = BY - Y4 - C*(BX-X4)
725.             DN(3) = S*YD
726.             YD = BY - Y1 + T*(BX - X1)
727.             DN(4) = CO*YD
728.             GO TO 80
729.          75  CONTINUE
730.             DN(1) = BX - X1
731.             DN(2) = Y2 - BY
```



```
732.          DN(3) = X3 - BX
733.          DN(4) = Y3 - Y1
734.      80  CONTINUE
735.          TOT = 0.0
736.          DO 83  N = 1,4
737.          IS(N) = 0
738.          D2(N) = DD-DN(N)
739.          IF (D2(N) .LE. 0.0)  GO TO 83
740.          TOT = TOT + D2(N)*D2(N)
741.      83  CONTINUE
742.          IF (TOT .GT. DD*DD)  GO TO 100
743.          RL = DD
744.          IRAD = 0
745.      81  CONTINUE
746.          AL = 3.14159*RL*PL
747.          DO 84  N = 1,4
748.          F(N) = 0.0
749.          R = DD - DN(N)
750.          IF ((R .GT. 0.0) .AND. (R .GE. RL))  F(N) = 1.0
751.          IF (F(N) .EQ. 1.0)  GO TO 84
752.          IF ((R .LT. 0.0) .AND. (-R .GT. RL))  GO TO 84
753.          IF (R .LT. 0.0)  R = -R
754.          IF (R .EQ. 0.0)  GO TO 82
755.          Z = R/RL
756.          RHO = 2.*ATAN((1./(Z*Z)-1.)**.5))
757.          A = RL*RL*(RHO - SIN(RHO))/2.
758.          IF (DN(N) .GT. DD)  A = AL - A
759.          F(N) = 1. - A/AL
760.          GO TO 84
761.      82  F(N) = .5
762.      84  CONTINUE
763.          FX = 1. - F(1) - F(3)
764.          FY = 1. - F(2) - F(4)
765.          -IF (FX .LT. 0.0)  FX = 0.0
766.          IF (FY .LT. 0.0)  FY = 0.0
767.          IF (KTEST .GT. 4)  WRITE (6,102)  FX, FY
768.      C  NOTE THAT THE USE OF FX AND FY PROVIDES ONLY AN APPROXIMATE RESULT
769.          FAC = FX*FY*AL/(TGT(L,12)*TGT(L,13))
770.          PS = 1. - AMIN1(1., FAC)
771.          IF (IRAD .EQ. 1)  GO TO 91
772.      90  P(L,1) = 1. - (1.-P(L,1))*PS
773.          IF (PP .LE. 1)  GO TO 92
774.          IRAD = 1
775.          RL = PP
776.          GO TO 81
777.      91  P(L,3) = 1. - (1. - P(L,3))*PS
778.          GO TO 93
779.      92  IF (PP .EQ. 0.0)  GO TO 93
780.          P(L,3) = 1. - (1. -P(L,3))*(1. - PP)
781.      93  CONTINUE
782.          IF (KTEST .GT. 3)  WRITE (6,1001)  L, P(L,1), P(L,3)
783.          IF (TGT(L,11) .LT. 1.)  GO TO 100
784.          IF (NHIT(L) .GT. 25)  GO TO 100
785.          DO 95  J = 1, NST
786.          IF (MHIT(J) .EQ. 0)  GO TO 100
787.          IF (MHIT(J) .NE. L)  GO TO 95
788.          HIT (J,1,NHIT(L))=BX
789.          HIT (J,2,NHIT(L))=BY
790.          HIT (J,3,NHIT(L))=ATT(1,9)
791.          GO TO 100
792.      95  CONTINUE
```

```
793. 100 CONTINUE
794. IF (NTGT .NE. 1) RETURN
795. 110 CONTINUE
796. DO 120 J = 1,5
797. IF (NRW(J) .EQ. 0) GO TO 130
798. IF (NRW(J) .NE. L) GO TO 120
799. HITR(J,1,NHIT(1)) = BX
800. HITR(J,2,NHIT(L)) = BY
801. HITR(J,3,NHIT(L)) = ATT(1,9)
802. 120 CONTINUE
803. 130 CONTINUE
804. RETURN
805. 101 FORMAT(' ',15X,'*** HITTGT NHIT(',I3,')',2F6.0)
806. 102 FORMAT(' ', ' FX ',F6.3,' FY ',F6.3)
807. 1001 FORMAT(' ', ' TGT # ',I3,' PK1 ',F8.3,' PK3 ',F8.3)
808. END
809. SUBROUTINE GAUSS (S,V)
810. COMMON/INT/NT,NA,ND,NTM,KTEST,MCR,MCW,MCL,MODE,NPRINT,NAM,NST,MT
811. IF ((KTEST .GT. 7) .OR. (MODE .LT. 0)) GO TO 50
812. A=0.0
813. C IF (MODE .EQ. 0) GO TO 20
814. C DO 10 I=1,12
815. C Y = RAND(1.)
816. C 10 A = A + Y
817. C GO TO 40
818. 20 DO 30 I=1,12
819. Y=AN(I)
820. 30 A = A + Y
821. 40 V = (A-6.0)*S
822. RETURN
823. 50 CONTINUE
824. V = 0.0
825. RETURN
826. END
827. SUBROUTINE CHECKR
828. COMMON /ARRAYS/ TGT(250,13), ATT(50,11),AMD(10,20,2), TD(250,2),
829. XIZONE(50,2),NHIT(250),MHIT(20),HIT(20,3,25),NRW(5),HITR(5,3,250)
830. X ,P(250,3),COV(250),MTYPE(10),NAME(250,2),WPNREL(10),NCRU(250)
831. COMMON/INT/NT,NA,ND,NTM,KTEST,MCR,MCW,MCL,MODE,NPRINT,NAM,NST,MT
832. COMMON/STATS/NTRIAL,ITRIAL,NSTAT,STAT(250,4),STAT2(20,5),MSTAT(8)
833. COMMON /CONTRL/ NREP,NPLLT,INW,INL,NSAVE1,NSAVE2,LIST,NJMEM
834. NC=0
835. NN=0
836. LHGLFS = 10000
837. DO 40 MRW = 1, 5
838. C CYCLE THRU AS MANY AS 8 RUNWAY/TAXIWAYS.
839. IRW = NRW(MRW)
840. C EXIT IF NO TARGET NUMBER (IRW) FOUND.
841. IF (IRW .EQ. 0) GO TO 50
842. NN=NN+1
843. IF (NHIT(IRW) .EQ. 0) GO TO 40
844. IF (KTEST .GT. 4) WRITE (6,102) IRW
845. INDEX = MRW
846. CALL RUNWAY (INDEX, IRW, ICOND, NHOLES)
847. C RUNWAY SUBROUTINE RETURNS ICOND = 0 IF RUNWAY HAS REQUIRED SPACE;
848. C ICOND = 1 IF NOT.
849. C
850. IF (ICOND .EQ. 1) NC=NC+1
851. IF (NHOLES .LT. LHGLFS) LHGLFS = NHOLES
852. IF (NPRINT .EQ. 4) WRITE(6,103)ITRIAL,IRW,NHIT(IRW),NHOLES
853. 40 CONTINUE
```

```
854.      50  IF (NC .EQ. NN) GO TO 60
855.      IF (NPRINT .LT. 3) WRITE (6,101)
856.      NSTAT = NSTAT + 1
857.      60  CONTINUE
858.      IF ((NPEP .EQ. 1) .AND. (NC .EQ. NN)) GO TO 70
859.      RETURN
860.      70  MSTAT(1) = MSTAT(1) + LHLES
861.      MSTAT(2) = MSTAT(2) + LHLES*LHOLES
862.      RETURN
863.      101 FORMAT(' ', ' AT LEAST ONE RUNWAY IS AVAILABLE')
864.      102 FORMAT(' ', ' CHECK TARGET #', I3)
865.      103 FORMAT(' ', 'TRIAL', I4, ' TGT', I4, ' HITS', I4, ' REPAIRS', I3)
866.      END
867.      SUBROUTINE RUNWAY (MRW, IRW, ICOND, NHOLES)
868.      COMMON /ARRAYS/ TGT(250,13), ATT(50,11), AMD(10,20,2), TO(250,2),
869.      XIZONE(50,2), NHIT(250), MHIT(20), HIT(20,3,25), NRW(5), HITR(5,3,250)
870.      X , P(250,3), COV(250), MTYPE(10), NAME(250,2), WPNREL(10), NCBU(250)
871.      COMMON /STATS/ NTRIAL, ITPIAL, NSTAT, STAT(250,8), STAT2(20,5), MSTAT(8)
872.      COMMON /INT/ NT, NA, ND, NTM, KTEST, MCR, MCW, MCL, MODE, NPRINT, NAM, NST, MTT
873.      COMMON /HITS/ XN(250), YN(250), NZ(250)
874.      COMMON /CONTRL/ NPLP, NPLC, INW, INL, NSAVE1, NSAVE2, LIST, NJMEM
875.      DIMENSION NTEST(250), YH(250,2)
876.      C
877.      C CHECKS FOR THE EXISTANCE OF THE SPECIFIED RUNWAY MINIMUMS (MCL X
878.      C MCW) ON EACH RUNWAY AND DESIGNATED TAXIWAY (TYPE #1 TARGETS),
879.      C STOPS SEARCHING A GIVEN RUNWAY WHENEVER REQUIREMENT IS SATISFIED.
880.      C
881.      TH=TGT(IPW,9)/57.3
882.      C
883.      C ESTABLISH ORIGIN (X0, Y0) FOR A RECTANGULAR COORDINATE SYSTEM WITH
884.      C THE X-AXIS ON THE MORE SOUTHERLY EDGE OF THE RUNWAY.
885.      C
886.      NHI = 0
887.      NHOLES = 1000
888.      DO 5 N = 1, 250
889.      5  NZ(N) = 0
890.      IF (TGT(IRW,12) .GT. TGT(IRW,13)) GO TO 10
891.      NDIR=1
892.      X0=TGT(IPW,1)
893.      Y0=TGT(IRW,2)
894.      LTH = TGT(IRW,13)
895.      WID = TGT(IRW,12)
896.      GO TO 20
897.      10  NDIR=2
898.      X0=TGT(IPW,7)
899.      Y0=TGT(IRW,8)
900.      LTH = TGT(IRW,12)
901.      WID = TGT(IRW,13)
902.      20  CONTINUE
903.      IF (KTEST .GT. 4) WRITE (6,1004) IRW, X0, Y0, LTH, WID, MCL, MCW
904.      IF (MCW .GT. WID) GO TO 320
905.      NHIT1 = NHIT(IRW)
906.      DO 50 J = 1, NHIT1
907.      IF (AMD(HITR(MRW,3,I),1,1) .LT. 0.) GO TO 50
908.      NTW = HITR(MRW,3,I)
909.      GO TO 60
910.      50  CONTINUE
911.      60  CONTINUE
912.      NON = 1
913.      EMD = AMD(NTW,1,1)
914.      DO 70 I = 1, NHIT1
```

```
915.          IF (HITR(MRW,3,1) .EQ. NTH) GO TO 70
916.          IF (AMD(HITR(MRW,3,1),1,1) .LT. 0.) GO TO 70
917.          NH = 0
918.          EMD = 0.0
919.          GO TO 80
920.          70 CONTINUE
921.          80 CONTINUE
922.          DO 140 I = 1, NHIT1
923.          C TRANSFORM HIT COORDINATES TO RUNWAY COORDINATES.
924.          NH = NH + 1
925.          XB = HITR(MRW,1,I)
926.          YB = HITR(MRW,2,I)
927.          IF (AMD(HITR(MRW,3,I),1,1) .LT. 0.) NZ(I) = 1
928.          IF (TH .EQ. 0.0) GO TO 110
929.          XX=XB-X0
930.          YY=YB-Y0
931.          R = (XX*XX+YY*YY)**(.5)
932.          IF (KTEST .GT. 7) WRITE (6,1010) R,XX,YY
933.          YZ = YY/R
934.          TH1 = ATAN(YY/XX)
935.          IF (XX .LT. 0.0) TH1 = TH1 + 3.1415
936.          TH2 = TH1 + TH
937.          IF (NTH .EQ. 2) TH2 = TH2 - 1.5706
938.          XN(I)=R*COS(TH2)
939.          YN(I)=R*SIN(TH2)
940.          IF (KTEST .GT. 5) WRITE (6,1009) I, XN(I), YN(I)
941.          GO TO 130
942.          110 IF (NDR .EQ. 2) GO TO 120
943.          XN(I)=XB-X0
944.          YN(I)=YB-Y0
945.          GO TO 130
946.          120 XN(I)=YB-Y0
947.          YN(I)=X0-XB
948.          130 IF ( I .GT. 249) GO TO 150
949.          140 CONTINUE
950.          GO TO 160
951.          150 WRITE (6,1001) IRW, ITRIAL
952.          160 CONTINUE
953.          NH = NHI
954.          C IF NPLOT .EQ. 2 RUNWAY IMPACTS ARE PLOTTED FOR ALL CONDITIONS.
955.          C IF NPLOT .EQ. 1 IMPACTS ONLY PLOTTED WHEN RUNWAY IS CLOSED.
956.          IF ((MRW .EQ. 1) .AND. (NPLOT .EQ. 2)) WRITE(6,1012) ITRIAL
957.          IF (NPLOT .EQ. 2) CALL PLOTHT(NH,IRW,LTH,WID)
958.          IF (KTEST .GT. 6) WRITE (6,1006) NHIT(IRW),NH
959.          DO 170 I=1,NH
960.          YH(I,1) = YN(I)
961.          170 YH(I,2)=I
962.          IF (NH .EQ. 1) GO TO 190
963.          DO 180 J=2,NH
964.          DO 180 K=2,NH
965.          C ORDER ALL HITS FROM LOWEST Y TO HIGHEST. YH(I,1) IS THE
966.          C Y COORDINATE, YH(I,2) THE HIT NUMBER, OF THE I TH ORDERED HIT.
967.          I=NH-K+2
968.          IF (YH(I,1) .GE. YH(I-1,1)) GO TO 180
969.          T=YH(I-1,1)
970.          TN=YH(I-1,2)
971.          YH(I-1,1)=YH(I,1)
972.          YH(I-1,2)=YH(I,2)
973.          YH(I,1)=T
974.          YH(I,2)=TN
975.          180 CONTINUE
```

```
976. 190 CONTINUE
977.    XL = 0.0
978.    XU = MCL
979. 200 YL = 0.0
980.    YU = MCL
981.    NYL = 1
982.    IF (NCN .EQ. 0) GO TO 210
983.    YL = YL - EMC
984.    YU = YU + EMC
985. 210 CONTINUE
986. 220 IHOLE = 0
987.    DO 250 I = NYL , NH
988.    IF (N2(I) .EQ. 1) GO TO 250
989.    YL = YH(I,1)
990.    IF (NON .EQ. 1) GO TO 230
991.    R = AMD(HITR(MRW,3,YH(I,2)),1,1)
992.    YL = YL - R
993.    YU = YU + R
994. 230 IF (YT .LT. YL) GO TO 240
995.    XT = XM(YH(I,2))
996.    IF ((XT .LT. XL) .OR. (XT .GT. XU)) GO TO 240
997.    IF (YT .GT. YU) GO TO 260
998.    IHOLE = IHOLE + 1
999.    IF (KTEST .GT. 6) WRITE(6,1008) I, IHOLE, YL, YU
1000.    NTEST(IHOLE) = I
1001.    IF (NREV .EQ. 0) GO TO 260
1002. 240 IF (NCN .EQ. 1) GO TO 250
1003.    IF (I .EQ. NH) GO TO 260
1004.    YL = YL + P
1005.    YU = YU - R
1006. 250 CONTINUE
1007. 260 CONTINUE
1008.    IF (IHOLE .EQ. 0) GO TO 300
1009.    IF (NCN .EQ. 1) NYL = NTEST(I)
1010.    NHOLE = MIN0(NHOLES, IHOLE)
1011.    IF (NON .EQ. 0) GO TO 270
1012.    YL = YL + INW
1013.    YU = YU + INW
1014.    IF (YU .GT. (WID+EMD)) GO TO 280
1015.    GO TO 220
1016. 270 YL = YL + R + INW
1017.    YU = YU - R + INW
1018.    IF (YU .GT. WID) GO TO 280
1019.    IF (KTEST .GT. 4) WRITE(6,1008) NHOLE
1020.    GO TO 220
1021. 280 CONTINUE
1022.    XL = XL + INL
1023.    XU = XU + INL
1024.    IF (XU .GT. LTH) GO TO 290
1025.    GO TO 200
1026. 290 IF (NPRINT .GE. 3) GO TO 295
1027.    IF ((MRW .EQ. 1) .AND. (NPLOT .NE. 2)) WRITE (6,1012) ITRIAL
1028.    WRITE(6,1002) IRW
1029.    IF (NREP .GT. 0) WRITE (6,1011) NHOLE
1030.    IF (NPLOT .EQ. 1) CALL PLOTHT(NH,IRW,LTH,WID)
1031. 295 ICOND = 1
1032.    IF ((NPLOT .GT. 0) .AND. (NPRINT .LT. 3)) WRITE(6,1000)
1033.    RETURN
1034. 300 IF (NPRINT .GT. 2) GO TO 310
1035.    IF ((MRW .EQ. 1) .AND. (NPLOT .NE. 2)) WRITE (6,1012) ITRIAL
1036.    WRITE (6,1003) IRW
```

```
1037. 310 ICOND = 0
1038. IF ((NPLOT .EQ. 2) .AND. (NPKINT .LT. 3)) WRITE(6,1000)
1039. NHOLS = 0
1040. RETURN
1041. 320 WRITE(6,1005) IRW
1042. STOP
1043. 1000 FORMAT('1')
1044. 1001 FORMAT('0','ONLY FIRST 250 HITS TESTED FOR TARGET #',I4,
1045. X ' IN TRIAL #',I4,/,/)
1046. 1002 FORMAT('0','RUNWAY #',I3,' IS CLOSED',/)
1047. 1003 FORMAT('0','RUNWAY #',I3,' IS OPEN',/)
1048. 1004 FORMAT(' ',' RUNWAY SPECS ',I4,4X,2F8.0,I8,F8.0,2I8)
1049. 1005 FORMAT('0','***** TARGET #',I3,' IS TOO NARROW FOR ',
1050. X ' FLIGHT OPERATIONS')
1051. 1006 FORMAT(' ',' # HITS TO CHECK', 2I6)
1052. 1008 FORMAT(' ',' TEST POINT #B ',2I10,2F10.0)
1053. 1009 FORMAT(' ',' TEST POINT #E', 14,2F10.0)
1054. 1010 FORMAT(' ',' R = ',F8.0,10X,2F10.0)
1055. 1011 FORMAT(' ',' I4,' HOLS MUST BE REPAIRED TO MEET RUNWAY ',
1056. X ' MINIMUMS',/)
1057. 1012 FORMAT('1',' ***** TRIAL #',I3,' *****',/)
1058. END
1059. SUBROUTINE PRINT
1060. INTEGER *4 NAME
1061. COMMON /ARPA5/ TGT(250,13), ATT(50,11), AMC(10,20,2), TO(250,2),
1062. XIZONE(50,2),NHIT(250),MHIT(20),HIT(20,3,25),NKW(5),HITR(5,3,250)
1063. X ,P(250,3),COV(250),MTYPE(10),NAME(250,2),WPNREL(10),NCBU(250)
1064. COMMON/INT/NT,NA,ND,NTM,KTEST,MCR,MCW,MCL,MODE,NPRINT,NAM,NST,MTT
1065. COMMON/STATS/NTRIAL,ITRIAL,NSTAT,STAT(250,8),STAT2(20,5),MSTAT(8)
1066. IF (NTRIAL .EQ. 1) GO TO 1
1067. WRITE (6,108) ITRIAL
1068. GO TO 3
1069. 1 CONTINUE
1070. WRITE (6,106)
1071. 3 CONTINUE
1072. WRITE (6,101)
1073. DO 10 M=1,MTT
1074. NN= 0
1075. DO 10 I=1,NT
1076. IF (TGT(I,10) .NE. M) GO TO 10
1077. NN= NN+1
1078. IF (NN .EQ. 1) WRITE (6,109) M
1079. NAID = NHIT(I) - NCBU(I)
1080. WRITE(6,102) I, NAID,COV(I),P(I,1),P(I,3),P(I,2),
1081. X NAME(I,1), NAME(I,2)
1082. 10 CONTINUE
1083. IF (NPRINT .GT. 0) GO TO 30
1084. WRITE (6,103)
1085. DO 20 M = 1, NST
1086. IF (MHIT(M) .EQ. 0) GO TO 30
1087. NN=0
1088. NL = NHIT(MHIT(M))
1089. IF (NL .EQ. 0) GO TO 20
1090. DO 15 N=1, NL
1091. NN=NN+1
1092. IF (NN .EQ. 26) GO TO 20
1093. IF (NN .EQ. 1) WRITE (6,104) MHIT(M)
1094. X=HIT(M,1,NN)
1095. Y=HIT(M,2,NN)
1096. NWPN=HIT(M,3,NN)
1097. WRITE (6,105) X,Y,NWPN
```

```

1098.      15  CONTINUE
1099.      20  CONTINUE
1100.      30  CONTINUE
1101.      DO 50  M = 1, 5
1102.      IF (NFW(M) .EQ. 0) GO TO 60
1103.      NN = 0
1104.      NL = NHIT(NRW(M))
1105.      IF (NL .EQ. 0) GO TO 50
1106.      DO 4)  N = 1, NL
1107.      NN = NN + 1
1108.      IF (NN .EQ. 251) GO TO 50
1109.      IF (NN .EQ. 1) WRITE (6,104)  NRW(M)
1110.      X = HITR(M,1,NN)
1111.      Y = HITR(M,2,NN)
1112.      NWPN = HITR(M,3,NN)
1113.      WRITE (6,105) X, Y, NWPN
1114.      40  CONTINUE
1115.      50  CONTINUE
1116.      60  CONTINUE
1117.      RETURN
1118.      101  FOPMAT('0',
1119.      X'TGT  NO.      CBU      ',10X,
1120.      X'  BOMBS      CBU      BLDG',/,1X,
1121.      X'NO.  HITS  COVERAGE ',10X,
1122.      X'EMD  OTHER      PK      NO.')
1123.      102  FORMAT(' ',13,3X,14,4X,F6.2,10X,2(3X,F5.3),6X,F5.3,7X,2A4)
1124.      103  FORMAT('0',15X,'HIT LOCATION AND WPN TYPE FOR SELECTED TARGETS'
1125.      X,/,/)
1126.      104  FORMAT(' ', ' TARGET NUMBER',14,' X-DIM  Y-DIM  ',
1127.      X 'WPN TYPE',/)
1128.      105  FORMAT(' ', 21X,2F9.0,17)
1129.      106  FORMAT('1',20X,'TARGET HIT SUMMARY',/)
1130.      108  FORMAT('1',10X,'TARGET HIT SUMMARY  TRIAL',15)
1131.      109  FORMAT('0', 10X,'** TARGET TYPE # ',13,' **',/)
1132.      END
1133.      SUBROUTINE STATIS
1134.      COMMON /ARRAYS/ TGT(250,13), ATT(50,11),AMD(10,20,2), TO(250,2),
1135.      XIZONE(50,2),NHIT(250),MHIT(20),HIT(20,3,25),NRW(5),HITR(5,3,250)
1136.      X ,PI(250,3),COV(250),MTYPE(10),NAME(250,2),WPNREL(10),NCRU(250)
1137.      COMMON/INT/NT,NA,ND,NTM,KTEST,MCR,NCW,MCL,MODE,NPRINT,NAM,NST,MTT
1138.      COMMON/STATS/NTRIAL, ITRIAL,NSTAT,STAT(250,8),STAT2(20,5),MSTAT(8)
1139.      COMMON /CONTRL/ NREP,NPLOT,INW,INL,NSAVE1,NSAVE2,LIST,NJMEN
1140.      WRITE (6,101) NTRIAL
1141.      AVGREP = 0.0
1142.      DO 12  M = 1, MTT
1143.      NN = 0
1144.      SUM1 = 0.0
1145.      SUM2 = 0.0
1146.      DO 10  I = 1, NT
1147.      IF (TGT(I,10) .NE. M) GO TO 10
1148.      CONTINUE
1149.      NN = NN + 1
1150.      IF (NN .EQ. 1) WRITE (6,102) M
1151.      FHIT = (STAT(I,3)/NTRIAL)*100.
1152.      AHITS= STAT(I,1)/NTRIAL
1153.      SUM1 = SUM1 + AHITS
1154.      TRIAL = NTRIAL
1155.      SDH = (STAT(I,2) - TRIAL*AHITS*AHITS)/(TRIAL - 1.)
1156.      SDH = SDH**(0.5)
1157.      ACOV = STAT(I,4)/TRIAL
1158.      SUM2 = SUM2 + ACOV

```

```
1159.          SDC = (STAT(1,5) - TRIAL*ACOV*ACOV)/(TRIAL - 1.)
1160.          SDC = SDC**(0.5)
1161.          PK1 = STAT(1,6)/TRIAL
1162.          PK2 = STAT(1,7)/TRIAL
1163.          PK3 = STAT(1,8)/TRIAL
1164.          WRITE (6,107) I, FHIT, AHITS, SDH, ACOV, SDC, PK1, PK3, PK2, NAME(1,1),
1165.          X   NAME(1,2)
1166.          10  CONTINUE
1167.          IF (NN .GT. 0)   WRITE (6,108)   SUM1, SUM2
1168.          12  CONTINUE
1169.          WRITE (6,106)
1170.          DO 25  M = 1,MTT
1171.          DO 15  I = 1,NT
1172.          IF ( TGT(1,10) .EQ. M)   GO TO 20
1173.          15  CONTINUE
1174.          GO TO 25
1175.          20  FHIT = STAT2(M,1)/TRIAL
1176.          SDH = (STAT2(M,2) - TRIAL*FHIT*FHIT)/(TRIAL - 1.)
1177.          FCCV3 = 100.*(STAT2(M,3)/TRIAL)
1178.          FCCV4 = 100.*(STAT2(M,4)/TRIAL)
1179.          FCCV5 = 100.*(STAT2(M,5)/TRIAL)
1180.          FHIT = 100.*FHIT
1181.          SDH = 100.*(SDH**(0.5))
1182.          WRITE (6,107) M, FHIT, SDH, FCCV3, FCCV5, FCCV4
1183.          25  CONTINUE
1184.          30  CONTINUE
1185.          IF (MCR .EQ. 0)   GO TO 50
1186.          STA = MSTAT
1187.          FOPEN = (STA/TRIAL)*100.
1188.          NCLSD = NTRIAL - STA
1189.          IF (NCLSD .EQ. 0)   GO TO 40
1190.          CLSD = NCLSD
1191.          AVGREP = MSTAT(1)/CLSD
1192.          SDREP = 0.0
1193.          IF (NCLSD .GT. 1)
1194.          X   SDREP = ((MSTAT(2)-CLSD*AVGREP*AVGREP)/(CLSD-1.))**.5)
1195.          40  CONTINUE
1196.          WRITE (6,104) FOPEN
1197.          IF (NREP .EQ. 1)   WRITE (6,105) AVGREP, SDREP
1198.          50  CONTINUE
1199.          RETURN
1200.          101  FORMAT ('1', 10X, 'TARGET DAMAGE STATISTICS FOR', I4, ' TRIALS',/,/,
1201.          X   ' TARGET   PERCENT   AVERAGE HITS   STD. DEV.   AVG. CBU STD.'
1202.          X   ', DEV.   AVG. 90MB COVERAGE CBU   BLDG',/,
1203.          X   ' NUMBER  ATTACKS HIT PER ATTACK   OF HITS   COVERAGE COVE',
1204.          X   'RAGE   EMD   OTHER   PK   NO.',/,/)
1205.          102  FORMAT (' ',/, 15X, 'TARGET TYPE # ', I3,/)
1206.          103  FORMAT(' ', I6,6X, F6.1,6X, F7.2,6X, F6.2,4X, F6.2,4X, F6.2,
1207.          X   2F9.3,3X, F9.3,2X, 2A4)
1208.          104  FORMAT(' ',/, ' AT LEAST ONE MINIMUM RUNWAY SECTION WAS OPEN AFTE
1209.          XR', F6.1, ' PERCENT OF THE ATTACKS',/)
1210.          105  FORMAT(' ', 'WHEN ALL RUNWAYS WERE CLOSED, ', F4.1, '( ', F4.1,
1211.          X   ') HOLES REQUIRED REPAIR, ON THE AVERAGE, TO PROVIDE',
1212.          X   ' A MINIMUM RUNWAY',/)
1213.          106  FORMAT(' ',/,/, 10X, 'DAMAGE STATISTICS BY TARGET TYPE',/,/,
1214.          X   '   AVERAGE',/,
1215.          X   '   TARGET   PERCENT   STANDARD   ---- COVERAGE ----',/,
1216.          X   '   TYPE   HIT   DEVIATION   EMD   OTHER   CBU',/)
1217.          107  FORMAT(' ',5X, I2, 2(6X, F5.1), 3(2X, F5.1))
1218.          108  FORMAT(' ',25X, '-----', 16X, '-----',/, 26X, F6.2, 16X, F6.2)
1219.          END
```





```
1281.      DO 5   K = 1, 16
1282.      5   NCOV(J,K) = 0
1283.      16   CONTINUE
1284.          XX(1) = XB - (SL+CW)/2.
1285.          XX(2) = XX(1) +SL
1286.          XX(3) = XX(2) + CW
1287.          XX(4) = XX(1) + CW
1288.          YY(1) = YB + (SW-CL)/2.
1289.          YY(2) = YY(1) + CL
1290.          YY(3) = YY(2) - SW
1291.          YY(4) = YY(1) - SW
1292.          IF (INIT2 .GT. 0) GO TO 18
1293.          INIT2 = 1
1294.      C   FIND WESTERLY CORNER
1295.          ILX = 1
1296.          DO 15  NN = 1,4
1297.          15  IF (XX(NN) .LT. XX(ILX))  ILX = NN
1298.              IF (S .EQ. 1.0)  ILX = 4
1299.      C   RENUMBER CORNERS SO THAT CORNER #1 IS THE MOST WESTERN
1300.          IDIF = ILX - 1
1301.      C   COMPUTE AND ADJUST TAN AND COTAN AS REQUIRED
1302.          IF ((S .EQ. 0.0) .OR. (C .EQ. 0.0)) GO TO 28
1303.          IF ((ILX .EQ. 2) .OR. (ILX .EQ. 4)) GO TO 26
1304.          T = S/C
1305.          GO TO 27
1306.          26  T = -C/S
1307.          27  CT = 1./T
1308.          28  CONTINUE
1309.          15  DO 22  NN = 1,4
1310.              NEW = NN - IDIF
1311.              IF (NEW .LT. 1)  NEW = NEW + 4
1312.              X(NEW) = XX(NN)
1313.              Y(NEW) = YY(NN)
1314.          22  IF (KTFST .GT. 4) WRITE(6,1003)NEW,Y(NEW),Y(NEW)
1315.              IF (INIT1 .GT. 0) GO TO 31
1316.      C   CREATE A 16-POINT GRID ON TARGET - USE MORE POINTS FOR LARGE TGT
1317.          INIT1 = 1
1318.          NX0 = 8
1319.          NY0 = 8
1320.          IF (TGT(L,12) .GT. 250.)  NY0 = 16
1321.          IF (TGT(L,12) .GT. 1000.)  NY0 = 32
1322.          IF (TGT(L,13) .GT. 250.)  NX0 = 16
1323.          IF (TGT(L,13) .GT. 1000.)  NX0 = 32
1324.          NXT = NX0/2
1325.          NYT = NY0/2
1326.          DO 29  J = 1, NXT
1327.          DO 29  K = 1, NYT
1328.              ICDV(J,K) = 0
1329.          29  PSCDV(J,K) = 1.0
1330.              NX1 = NX0 - 1
1331.              NY1 = NY0 - 1
1332.              X0 = NX0
1333.              Y0 = NY0
1334.              A0 = TGT(L,1)
1335.              A1 = (TGT(L,7)-TGT(L,1))/X0
1336.              A2 = (TGT(L,3)-TGT(L,1))/Y0
1337.              A3 = TGT(L,2)
1338.              A4 = (TGT(L,4)-TGT(L,2))/Y0
1339.              A5 = (TGT(L,8)-TGT(L,2))/X0
1340.          31  CONTINUE
1341.      C   TEST TO SEE IF TARGET CORNERS COVERED BY PATTERN
```

```
1342.         NIN = 0
1343.         NGIN = 0
1344.         DO 10      NC = 1, 7, 2
1345.         XT=TGT(L,NC)
1346.         YT=TGT(L,NC+1)
1347.         IF (KTEST .GT. 4) WRITE(6,1001) XT,YT
1348.         IF ((YT .GT. Y(1)).OR.(XT .GT. X(3))) GO TO 10
1349.         IF ((YT .GT. Y(4)).OR.(YT .GT. Y(2))) GO TO 10
1350.         IF ((X(1) .GT. X(2)).OR.(X(3) .GT. X(4))) GO TO 10
1351.         IF ((Y(1)+CT*(XT-X(1))) .GT. Y(1)) GO TO 10
1352.         IF ((Y(2)-T*(XT-X(2))) .GT. Y(2)) GO TO 10
1353.         IF ((Y(4)+CT*(XT-X(4))) .GT. Y(4)) GO TO 10
1354.         IF ((Y(1)-T*(XT-X(1))) .GT. Y(1)) GO TO 10
1355.         9  NIN = NIN + 1
1356.         10 CONTINUE
1357.         IF (KTEST .GT. 3) WRITE(6,1002) NIN
1358.         IF (NIN .LT. 4) GO TO 34
1359.         C   IF ALL CORNERS COVERED BY PATTERN, TARGET FULLY COVERED
1360.         DO 32      J = 1, NXT
1361.         DO 32      K = 1, NYT
1362.         32  NCOV(J,K) = NCOV(J,K) + 1
1363.         GO TO 33
1364.         C   IF PARTIALLY COVERED, ESTIMATE FRACTION THAT IS COVERED
1365.         34  CONTINUE
1366.         DO 30      NX = 1,NX1,2
1367.         DO 30      NY = 1,NY1,2
1368.         J = (NX+1)/2.
1369.         K = (NY+1)/2.
1370.         C   GRID-POINT DIMENSIONS
1371.         XT = AC + NX*A1 + NY*A2
1372.         YT = AS + NY*A4 + NX*A5
1373.         C   CHECK IF WITHIN RECTANGLE ENCLOSED BY PATTERN THAT IS PARALLEL TO
1374.         C   AXES
1375.         IF ((XT .LT. X(1)).OR.(XT .GT. X(3))) GO TO 30
1376.         IF ((YT .LT. Y(4)).OR.(YT .GT. Y(2))) GO TO 30
1377.         IF ((S .EQ. 1.) .OR. (C .EQ. 0.)) GO TO 35
1378.         C   CHECK IF POINT IS WITHIN ACTUAL CBU PATTERN
1379.         IF (YT .GT. (Y(1)+CT*(XT-X(1)))) GO TO 30
1380.         IF (YT .GT. (Y(2)-T*(XT-X(2)))) GO TO 30
1381.         IF (YT .LT. (Y(4)+CT*(XT-X(4)))) GO TO 30
1382.         IF (YT .LT. (Y(1)-T*(XT-X(1)))) GO TO 30
1383.         35  NGIN = NGIN + 1
1384.         NCOV(J,K) = NCOV(J,K) + 1
1385.         IF (KTEST .GT. 5) WRITE(6,1005) NX,NY,XT,YT,NGIN,NCOV(J,K)
1386.         30  CONTINUE
1387.         33  CONTINUE
1388.         IF ((NIN + NGIN) .EQ. 0) GO TO 20
1389.         C   RECORD ANY COVERAGE AS A 'HIT'
1390.         NHIT(L) = NHIT(L) + 1
1391.         NCBU(L) = NCBU(L) + 1
1392.         IF (KTEST .GT. 4) WRITE(6,1006) L, NHIT(L), NCBU(L)
1393.         IF ((TGT(L,11) .LT. 1) .OR. (TGT(L,10) .EQ. 1.)) GO TO 130
1394.         IF (NHIT(L) .GT. 25) GO TO 130
1395.         DO 120      J = 1, NST
1396.         IF (NHIT(J) .EQ. 0) GO TO 130
1397.         IF (NHIT(J) .NE. L) GO TO 120
1398.         HIT (J,1,NHIT(L))=XB
1399.         HIT (J,2,NHIT(L))=YB
1400.         HIT (J,3,NHIT(L))=NWPB
1401.         IF (NHIT(L) .EQ. 25) WRITE (6,1007) L, ITRIAL
1402.         GO TO 130
```

```

1403. 120 CONTINUE
1404. 130 CONTINUE
1405. IF (TGT(L,10) .NE. 1.) GO TO 150
1406. IF (NHIT(L) .GT. 250) GO TO 150
1407. DO 140 J = 1, 5
1408. IF (NRW(J) .EQ. 0) GO TO 150
1409. IF (NRW(J) .NE. 1) GO TO 140
1410. HITR(J,1,NHIT(L)) = X3
1411. HITR(J,2,NHIT(L)) = Y8
1412. HITR(J,3,NHIT(L)) = NWPN
1413. IF (NHIT(L) .EQ. 250) WRITE(6,1009) L, ITRIAL
1414. 140 CONTINUE
1415. 150 CONTINUE
1416. C ***** RECYCLE FOR MORE WEAPONS
1417. 20 CONTINUE
1418. IF (INIT2 .EQ. 0) GO TO 40
1419. PSP = 1. - AMO(NWPN,TGT(L,10),2)
1420. DO 160 J = 1, NXT
1421. DO 160 K = 1, NYT
1422. ICOV(J,K) = ICOV(J,K) + NCOV(J,K)
1423. 160 PSCOV(J,K) = PSCOV(J,K)*PSP**NCOV(J,K)
1424. C ***** RECYCLE FOR MORE ATTACKS
1425. 40 CONTINUE
1426. IF ((INIT1+INIT2) .GT. 0) GO TO 165
1427. P(L,2) = 0.0
1428. COV(L) = 0.0
1429. GO TO 200
1430. 165 TCOV = 0.0
1431. PST = 0.0
1432. DO 170 J = 1, NXT
1433. DO 170 K = 1, NYT
1434. IF (ICOV(J,K) .GT. 0) TCOV = TCOV + 1.
1435. 170 PST = PST + PSCOV(J,K)
1436. TOTC = NXT*NYT
1437. COV(L) = TCOV/TOTC
1438. P(L,2) = 1. - PST/TOTC
1439. IF (KTEST .GT. 3)WRITE(6,1004)L,TCOV,TOTC,PST,COV(L),P(L,2)
1440. C ***** RECYCLE FOR MORE TARGETS
1441. 200 CONTINUE
1442. RETURN
1443. 1001 FORMAT(' ', ' XT ',F8.0,' YT ',F8.0)
1444. 1002 FORMAT(' ', ' NIN ',I4)
1445. 1003 FORMAT(' ', ' NEW ',I3,2F10.0)
1446. 1004 FORMAT(' ', ' TGT ',I4,' COV ',F6.0,' TOT ',F6.0,' PST ',
1447. X F10.4,' COV ',F10.4,' P(L,2) ',F10.4)
1448. 1005 FORMAT(' ', ' NX ',I3,' NY ',I3,2F10.0,2I10)
1449. 1006 FORMAT(' ', ' TGT ',I4,' NHI ',I5,' NCBU ',I5)
1450. 1007 FORMAT('0','***** ONLY 25 HITS WERE STORED FOR TARGET #',
1451. X I3,' DURING TRIAL #',I4,' *****')
1452. 1009 FORMAT('0','***** ONLY 250 HITS WERE STORED FOR TARGET ',
1453. X ' #',I3,' DURING TRIAL #',I4,' *****')
1454. END
1455. SUBROUTINE PLOTHT(NH,NR,LTH,WID)
1456. COMMON /HITS/ XN(250),YN(250),NZ(250)
1457. DIMENSION ICOL(130)
1458. DATA IBK / 1H /,IX/ 1H* /,IY/ 1H+ /,IS/ 1H- /,IE/ 1H' /
1459. C THIS ROUTINE PLOTS THE IMPACT POINTS (BUT NOT CRATERS) FOR
1460. C ALL HITS THAT HAVE BEEN STORED FOR A RUNWAY/TAXIWAY. IT
1461. C WILL PLOT ALL HITS THAT AFFECT RUNWAY OPERATION UP TO 50 'FEET'
1462. C OF EITHER SIDE OF (UP TO) A 300 'FOOT' RUNWAY. RUNWAY LENGTH
1463. C IS LIMITED TO 13000 'FEET'.

```

```
1464.          IWID = NID/10. + 5.
1465.          LFN = LTH/100. + 1.
1466.          IF (LEN .GT. 129) LEN = 129
1467.          LU = LFN/10.
1468.          LI = 10*LU + 1
1469.          LU = LU + 1
1470.          DO 40  J = 1,40
1471.              I = 41-J
1472.          DO 10  N = 1,129
1473.      10  ICCL(N) = 18K
1474.          ICCL(1) = IE
1475.          ICOL(LEN) = IF
1476.          IF ((I .NE. 5) .AND. (I .NE. IWID)) GO TO 14
1477.          DO 11  NS = 1,LFN
1478.      11  ICOL(NS) = IS
1479.          DO 12  NS = 1,LI,10
1480.      12  ICOL(NS) = IF
1481.      14  CONTINUE
1482.          DO 20  N = 1,NH
1483.          NY = YN(N)/10. + 5.
1484.          IF (NY .NE. 1) GO TO 20
1485.          NX = XN(N)/100. + 1.
1486.          IF ((NX .LT. 1) .OR. (NX .GT. 129)) GO TO 20
1487.          ICOL(NX) = IX
1488.          IF (NZ(N) .EQ. 1) ICOL(NX) = IY
1489.      20  CONTINUE
1490.          Y = I/5.
1491.          LY=Y
1492.          IF ((Y-LY) .NE. 0.0) GO TO 30
1493.          LYY = 5*LY - 5
1494.          WRITE (6,101) LYY,(ICOL(M),M=1,129)
1495.          GO TO 40
1496.      30  WRITE (6,102) (ICOL(M),M=1,129)
1497.      40  CONTINUE
1498.          WRITE (6,103) (I, I=1,12), NR
1499.          RETURN
1500.      101  FORMAT (' ',12,129A1)
1501.      102  FORMAT (' ',2X,129A1)
1502.      103  FORMAT (' ', ' 0', 12(8X,12) //
1503.          X 40X, ' TENS BY THOUSANDS OF LENGTH UNITS ',/,/,
1504.          X 40X, ' IMPACT POINTS ON RUNWAY NUMBER ',12,/,
1505.          X 37X, ' (1 = POINT IMPACT WPN * = CBU CENTROID)')
1506.          END
1507.          SUBROUTINE JMEMG(NJMEM,0)
1508.          COMMON/INT/NT,NA,ND,NTM,KTEST,MCR,MOW,NCL,MODE,NPRINT,NAM,IST,MTT
1509.          DIMENSION F(9), D(11)
1510.          C THIS SUBROUTINE PROVIDES THE USER METHOD OF AS OUTLINED IN THE
1511.          C 8-74 WANG 700 USERS'S MANUAL FOR JMEM OPEN-END METHODS.
1512.          C THIS TRAJECTORY PROGRAM PERMITS AIDA USERS TO PRESCRIBE THE
1513.          C ATTACK DATA AS IN JMEM. THE SUBROUTINE LOGIC IS TAKEN DIRECTLY FROM
1514.          C THE REFERENCED PUBLICATION AND USES NOTATION CLOSELY PARALLELING
1515.          C THE ORIGINAL. ONLY THE 'PATTERN RADIUS' COMPUTATION (USED WITH
1516.          C ROCKEYE) IS OMITTED.
1517.          NJMEM = NJMEM + 1
1518.          NCNT = 0
1519.          IF (NJMEM .EQ. 1) WRITE (6,101)
1520.          READ (6,102) (E(I),I=1,9)
1521.          WRITE (6,103) NA, (D(I),I=1,6)
1522.          WRITE (6,104) (E(I),I=1,9)
1523.          DO 20  I = 7,9
1524.      20  D(I) = D(I+1)
```

```
1525.          VI = 1.688*E(1)
1526.          V = VI
1527.          TH = E(3)/57.3
1528.          VX = V*COS(TH)
1529.          VY = -V*SIN(TH)
1530.          TR = (D(7) - 1.)*E(8) + (E(5)-1.)*E(9)
1531.          DX = TR*VX/2.
1532.          YI = E(2) - VY*TR
1533.          YI = (YI + E(2))/2.
1534.          Y = YI
1535.          TF = 0.0
1536.          VF = 0.0
1537.          THF=0.0
1538.          IF (:(7) .GT. 500.) GO TO 1
1539.          YI = 0.0
1540.          IF (:(7) .EQ. 0.0) GO TO 2
1541.          TPD=F(7)
1542.          GO TO 3
1543.          1  YI = F(7)
1544.          2  TPD= 99.
1545.          3  CONTINUE
1546.          DG = 32.17/(E(4)*F(4))
1547.          7  CONTINUE
1548.          IF (KTEST .GT. 5) WRITE (6,106) TF,Y,V
1549.          NCNT = NCNT + 1
1550.          IF (NCNT .GT. 2000) GO TO 30
1551.          NCNT2 = 0
1552.          C = V/220. - 3.
1553.          IF (C .GE. 3.) C = 3.
1554.          IF (C .LT. 1.) C = 1.
1555.          DD = C*DG*EXP(-Y/31000.)
1556.          DT = 10./(DG*V*V)
1557.          IF (DT .GT. 0.5) DT = 0.5
1558.          GO TO 13
1559.          5  CONTINUE
1560.          TF = TFD
1561.          VY = VY0
1562.          Y = Y0
1563.          IF (KTEST .GT. 5) WRITE(6,107) DT, VY
1564.          NCNT2 = NCNT2 + 1
1565.          IF (NCNT2 .GT. 100) GO TO 30
1566.          15 CONTINUE
1567.          TFD = TF
1568.          TF = TF + DT
1569.          IF (TF .LT. TPD) GO TO 4
1570.          TF = TPD
1571.          DT = TF - TFD
1572.          4  CONTINUE
1573.          VY0 = VY
1574.          VY = VY*(1.-V*DT*DD) - 32.17*DT
1575.          Y0 = Y
1576.          Y = Y + DT*(VY+VY0)/2.
1577.          IF ((Y - YI) .GE. -1.) GO TO 6
1578.          ZZ = VY0*V*DD + 32.17
1579.          Z = VY0*Y0 + 2.0*(Y0-YI)*ZZ
1580.          DT = (VY0 + Z**(.5))/ZZ
1581.          GO TO 5
1582.          6  CONTINUE
1583.          VX0 = VX
1584.          VX = VX*(1.-V*DT*DD)
1585.          DX = DX + DT*(VX0+VX)/2.
```

```
1586.      V = (VX*VX + VY*VY)**(.5)
1587.      IF (TF .EQ. TPD) GO TO 12
1588.      IF (E(7) .GT. 500.) GO TO 11
1589.      GO TO 10
1590.      11 AID = ABS((Y-E(7)))
1591.      IF (AID .LT. 1.) GO TO 12
1592.      IF (KTEST .GE. 5) WRITE (6,109) AID, TF
1593.      10 CONTINUE
1594.      IF (KTEST .GE. 5) WRITE(6,108) Y
1595.      IF (Y .GE. 1.) GO TO 7
1596.      GO TO 15
1597.      12 CONTINUE
1598.      Z = -VY/V
1599.      ZZ = (1.-Z*Z)**(.5)
1600.      THF=ATAN(Z/ZZ)
1601.      DG = 32.17/(E(5)*E(5))
1602.      TPD= 99.
1603.      YT = 0.0
1604.      IF (E(7) .GE. 500.) E(7) = 0.0
1605.      GO TO 10
1606.      15 Z = -VY/V
1607.      ZZ = (1.-Z*Z)**(.5)
1608.      AI = ATAN(Z/ZZ)
1609.      SR = (DX*DX + Y1*Y1)**(.5)
1610.      IF (D(5) .EQ. 0.0) GO TO 8
1611.      RP = D(4)/1000.
1612.      CP = D(5)/1000.
1613.      GO TO 9
1614.      8 BP = 0.000573*D(4)
1615.      CP = BP
1616.      9 CONTINUE
1617.      U = (E(6)*E(6)*TF*YF)
1618.      D(4)=(U+(SR*SR*RP/Y1)**2. )**(.5)
1619.      D(5)=(U+CP*CP*SR*SR)**(.5)
1620.      SRT = SR/1000.
1621.      D(8) = V1*TR*SIN(AI-TH)/SIN(AI)
1622.      DISP = 7(6)
1623.      D(10) = 0.674*DISP*SRT
1624.      D(6) = D(10)/Z
1625.      AID = 57.3*AI
1626.      THFD= 57.3*THF
1627.      WRITE(6,105) TF, SR, AID, THFD
1628.      RETURN
1629.      30 WRITE (6,110)
1630.      STOP
1631.      101 FORMAT('1',20X,'JMEM FCRMAT ATTACK DATA'/' ' ,
1632.      X'ATT NO.      HOG      X-MPI      Y-MPI      CEP(REP)      (DEP)      DIS
1633.      XP'/' ' ,
1634.      X'      SPEED      ALTITUDE      DIVE      TERM1      TERM2      WIN
1635.      XD      TD/HF      T      TD',/,/)
1636.      102 FORMAT( 6X, 6F6.0, 3F6.3)
1637.      103 FORMAT (10',16,7F10.0)
1638.      104 FORMAT (' ',6X, 6F10.0, 3F10.3)
1639.      105 FORMAT(' ',30X,'TF ',F5.2,' SEC SR ',F7.0,' FEET ',
1640.      X 'IMPACT ANGLE ',F5.2,' DEG (FUZING ANGLE ',F5.2,' DEG)')
1641.      106 FORMAT(' ', ' * ',F6.3,' SEC ',F8.0,' FEET ',F8.1,' FT/SEC')
1642.      107 FORMAT(' ',10X,' DT ',F10.4,' VERT VEL',F10.3)
1643.      108 FORMAT(' ', 'ALT ',F10.2)
1644.      109 FORMAT(' ', 'ALT DIFF',F10.2,' TF ',F10.2)
1645.      110 FORMAT('0'/' ', 'LOOPING IN JMEMO. CHECK INPUTS AND/OR TEST',
1646.      X ' WITH KTEST = 6.0')
```

```
1647.      END
1648.      SUBROUTINE EXPHIT
1649.      COMMON /ARPAYS/ TGT(250,13), ATT(50,11),AMD(10,20,2),T0(250,2),
1650.      XIZONE(50,2),NHIT(250),MHIT(20),HIT(20,3,25),NPW(5),HITR(5,3,250)
1651.      X ,P(250,3),COV(250),MTYPE(10),NAME(250,2),WPNREL(10),NCBU(250)
1652.      COMMON/INT/NT,NA,ND,NTM,KTEST,MCR,MCW,MCL,MODE,NPRINT,NAM,NST,MTT
1653.      COMMON / HITDN / NHITO, NRFD0
1654.      THIS ROUTINE ESTIMATES THE EXPECTED NUMBERS OF HITS FOR ATTACKS WITH
1655.      POINT-IMPACT WEAPONS ONLY.
1656.      C
1657.      C
1658.      C
1659.      NCYCLE = 0
1660.      PI = 3.14159
1661.      DO 200 I = 1,NA
1662.      NW = ATT(I,9)
1663.      IF (AMD(NW,1,1) .LT. 0.0) GO TO 240
1664.      REL = WPNREL(NW)
1665.      NFLAG = 0
1666.      NS = ATT(I,7)
1667.      LS = ATT(I,8)
1668.      PHI = ATT(I,1)/57.3
1669.      S = SIN(PHI)
1670.      C = COS(PHI)
1671.      IF ((ATT(I,1) .EQ. 0.) .OR. (ATT(I,1) .EQ. 180.)) GO TO 10
1672.      CT = C/S
1673.      GO TO 20
1674.      10 NFLAG = 1
1675.      20 CONTINUE
1676.      NP = 10
1677.      DLS = LS/9.
1678.      IF (LS .GT. 500) GO TO 30
1679.      NP = 5
1680.      DLS = LS/4.
1681.      IF (LS .GT. 50) GO TO 30
1682.      NP = 2
1683.      30 DLS = LS
1684.      CONTINUE
1685.      X = ATT(I,2) - 5*LS/2.
1686.      Y = ATT(I,3) - C*LS/2.
1687.      SIGRS = 2.200*ATT(I,4)*ATT(I,4) + ATT(I,6)*ATT(I,6)
1688.      SIGDS = 2.200*ATT(I,5)*ATT(I,5) + ATT(I,10)*ATT(I,10)
1689.      TSRS = 2. *SIGRS
1690.      TSDS = 2. *SIGDS
1691.      SIGR = SIGRS**(.5)
1692.      SIGD = SIGDS**(.5)
1693.      F = 1. / (SIGR*SIGD*6.2832)
1694.      IF (KTEST .GT. 1) WRITE(6,1008)I,LS,NP,DLS,X,Y,SIGR,
1695.      X SIGD, F
1696.      NFLAG2 = 0
1697.      DO 180 L = 1,NT
1698.      IF (TGT(L,10) .EQ. 21.) GO TO 180
1699.      NFLAG2 = 1
1700.      DEN = 0.0
1701.      TL1 = TGT(L,12)
1702.      TL2 = TGT(L,13)
1703.      TH = TGT(L,9)/57.3
1704.      CHI = TH - PHI
1705.      SC = SIN(CHI)
1706.      CC = COS(CHI)
1707.      S1 = (SIGRS*CC*CC + SIGDS*SC*SC)**(.5)
```



```
1708.      S2 = (SIGRS*SC*SC + SIGDS*CC*CC)**(.5)
1709.      N1 = (4.*TL1/S1) + 1
1710.      N2 = (4.*TL2/S2) + 1
1711.      IF (KTEST .GT. 1) WRITE(6,1009) N1, N2, S1, S2
1712.      DO 90   K = 1,7,2
1713.      XT = TGT(L,K)
1714.      YT = TGT(L,K+1)
1715.      XX = X - XT
1716.      IF (NFLAG .EQ. 1) GO TO 40
1717.      YY = YT - Y + CT*XX
1718.      D = -YY*S
1719.      R = YY*C - XX/S
1720.      GO TO 50
1721. 40     D = XX
1722.      R = (YT - Y)*C
1723. 50     AID1 = D*D/TSDS
1724.      IF ((TL1+TL2) .GT. 200.) GO TO 60
1725.      IF (AID1 .GT. 10.) GO TO 180
1726. 40     IF (AID1 .GT. 10.) GO TO 90
1727.      DD = SEXP(-AID1)
1728.      DO 80   M = 1,NP
1729.      AID2 = R*R/TSRS
1730.      IF (AID2 .GT. 10.) GO TO 70
1731.      DR = SEXP(-AID2)
1732.      DEN = DEN + DR*DD
1733.      NCYCLE = NCYCLE + 1
1734.      IF (KTEST .GT. 2) WRITE(6,1005) L,K,XT,YT,D,R,DD,DR,CEN
1735. 70     R = R - DLS
1736. 80     CONTINUE
1737. 90     CONTINUE
1738. C     IF THE TARGET DIMENSIONS ARE SMALL (I.E. LESS THAN ONE-QUARTER
1739. C     THE PROJECTION OF SIGMA PARALLEL TO THE TARGET EDGE) THE HIT
1740. C     DENSITY IS TAKEN AS THE AVERAGE OF THE VALUES AT THE FOUR CORNERS.
1741. C     IF IT IS LARGER, A GRID OF INTERNAL POINTS IS ESTABLISHED AND
1742. C     THE HIT DENSITY IS TAKEN AS THE AVERAGE OVER THE CORNERS AND THE
1743. C     INTERNAL POINTS.
1744.      IF ((N1 + N2) .GT. 2) GO TO 100
1745.      DEN = DEN/(4.*NP)
1746.      GO TO 160
1747. 100    CONTINUE
1748.      STG = SIN(TH)
1749.      CTG = COS(TH)
1750.      OIL1 = TL1/(N1+1)
1751.      OIL2 = TL2/(N2+1)
1752.      DO 150 M = 1,N1
1753.      DO 150 N = 1,N2
1754.      XT = TGT(L,1) + M*OIL1*STG + N*OIL2*CTG
1755.      YT = TGT(L,2) + M*OIL1*CTG - N*OIL2*STG
1756.      XX = X - XT
1757.      IF (NFLAG .EQ. 1) GO TO 110
1758.      YY = YT - Y + CT*XX
1759.      D = -YY*S
1760.      R = YY*C - XX/S
1761.      GO TO 120
1762. 110    D = XX
1763.      K = (YT - Y)*C
1764. 120    AID1 = D*D/TSDS
1765.      IF (AID1 .GT. 10.) GO TO 150
1766.      DD = SEXP(-AID1)
1767.      DO 140 K = 1,NP
1768.      AID2 = R*R/TSRS
```

```

1769.      IF (AID2 .GT. 10.) GO TO 130
1770.      DR = SEXP(-AID2)
1771.      DEN = DFN + DR*DC
1772.      NCYCLE = NCYCLF + 1
1773.      IF (KTEST .GT. 2) WRITE(6,1006) I,L,XT,YT,D R,DD,DR,DEN
1774.      130 R = R - DLS
1775.      140 CONTINUE
1776.      150 CONTINUE
1777.          DEN = DEN / ((4+N1*N2)*NP)
1778.      160 CONTINUE
1779.          DEN1 = DEN
1780.          DEN = NS*F*REL*ATT(I,11)*DEN1
1781.          IF (KTEST .GT. 1) WRITE (6,1007) I,L,DEN1,DEN
1782.          EMD = AMD*(NW,TGT(L,10),1)
1783.          DL1 = TL1 + 2*EMD
1784.          DL2 = TL2 + 2*EMD
1785.          EHT1 = DEN*(DL1*DL2 - .8584*EMD*EMD)
1786.          COV(L) = COV(L) + EHT1
1787.      180 CONTINUE
1788.          IF (NFLAG2 .EQ. 0) GO TO 230
1789.      200 CONTINUE
1790.          WRITE(6,1000) NCYCLF
1791.          WRITE (6,1002)
1792.          GO 220 M = 1,MTT
1793.          NN = 0
1794.          TCOV = 0.0
1795.          BLDGHT = 0.0
1796.          DO 210 L = 1,NT
1797.          IF (TGT(L,10) .NE. M) GO TO 210
1798.          NN = NN + 1
1799.          TCOV = TCOV + COV(L)
1800.          BLDGHT = BLDGHT + (1. - EXP(-COV(L)))
1801.          IF (NN .EQ. 1) WRITE (6,1001) M
1802.          WRITE (6,1003) L,COV(L),NAME(L,1),NAME(L,2)
1803.      210 CONTINUE
1804.          IF (NN .EQ. 0) GO TO 220
1805.          BLDGHT = BLDGHT/NN
1806.          WRITE (6,1010) TCOV, BLDGHT
1807.      220 CONTINUE
1808.      230 CONTINUE
1809.          RETURN
1810.      240 WRITE (6, 1004)
1811.          STOP
1812.      1000 FORMAT('1', 25X, 'CYCLES', I7//)
1813.      1001 FORMAT ('0',10X,'** TARGET TYPE #',I3,' **',/)
1814.      1002 FORMAT ('0',10X,'TARGET HITS',/,10X,' NO EXPECTED')
1815.      1003 FORMAT (' ',10X,I4,6X,F6.3,6X,2A4)
1816.      1004 FORMAT ('1', ' COMPUTATION STOPPED - CBU WEAPONS ARE NOT',
1817.      X ' PERMITTED WITH SUBROUTINE EXPHIT')
1818.      1005 FORMAT(' ','CD ATT ',I3,' TGT ',I3,' COR ',I2,
1819.      X 4F6.0, 3E12.4 )
1820.      1006 FORMAT (' ','IP ATT ',I3,' TGT ',I3,4F6.0,3E12.5)
1821.      1007 FORMAT(' ',' ATT ',I3,' TGT ',I3,' AVG DEN ',
1822.      X F8.5,' NOR DEN ',F14.10//)
1823.      1008 FORMAT (' ','ATT ',I3,2I8, F8.2, 4F8.1, E12.5)
1824.      1009 FORMAT(' ','N1 ',I4,' N2 ',I4,6X,2F10.1)
1825.      1010 FORMAT(' ',20X,'-----',/,19X,F8.3,'( ',F5.3,' )',,
1826.      END
1827.      FUNCTION SEXP(X)
1828.      IF (X .LT. -0.025) GO TO 10
1829.      SEXP = 1.+X

```

```
1830. RETURN
1831. 10 SEXP = EXP(X)
1832. RETURN
1833. END
1834. SUBROUTINE GRIDEN
1835. COMMON/ARRAYS/TGT(250,13),ATT(50,11),AMD(10,20,2),TO(250,2),
1836. X(250,2),NHIT(250),MHIT(20),HIT(20,3,25),NRW(5),HITR(5,3,250)
1837. X,P(250,3),CV(250),MYPE(10),NAME(250,2),WPNREL(10),NCHU(250)
1838. COMMON/INT/NT,NA,ND,NTM,KTEST,MCR,MCW,MCL,MODE,NPRINT,NAM,NST,MTT
1839. DIMENSION DATA(17)
1840. NCYCLE = 0
1841. PI = 3.14159
1842. SRPI = 1.7724
1843. DO 140 L = 1,NT
1844. IF (TGT(L,10) .NE. 21.) GO TO 140
1845. NFLAG2 = 0
1846. INC = 250
1847. IF (TGT(L,9) .GT. 0.) INC = TGT(L,9)/16.
1848. NX1 = TGT(L,1)
1849. NY1 = TGT(L,2)
1850. DO 120 NYR = 1,17
1851. DO 120 NX = 1,17
1852. NY = 18 - NYR
1853. DEN = 0
1854. XT = NX1 + (NX-1)*INC
1855. YT = NY1 + (NY-1)*INC
1856. DATA(NX) = 0.0
1857. DO 100 I = 1, NA
1858. NW = ATT(I,9)
1859. IF (AMD(NW,1,1) .LT. 0.) GO TO 160
1860. REL = WPNREL(NW)
1861. NFLAG = 0
1862. NS = ATT(I,7)
1863. LS = ATT(I,8)
1864. PHI = ATT(I,11)/57.3
1865. S = SIN(PHI)
1866. C = COS(PHI)
1867. IF ((ATT(I,1) .EQ. 0.) .OR. (ATT(I,1) .EQ. 130.)) GO TO 10
1868. CT = C/S
1869. GO TO 20
1870. 10 NFLAG = 1
1871. 20 CONTINUE
1872. NP = 10
1873. DLS = LS/9.
1874. IF (LS .GT. 500) GO TO 30
1875. NP = 5
1876. DLS = LS/4.
1877. IF (LS .GT. 50) GO TO 30
1878. NP = 2
1879. DLS = LS
1880. 30 CONTINUE
1881. X = ATT(I,2) -S*LS/2.
1882. Y = ATT(I,3) -C*LS/2.
1883. SIGRS = 2.200*ATT(I,4)*ATT(I,4) + ATT(I,6)*ATT(I,6)
1884. SIGDS = 2.200*ATT(I,5)*ATT(I,5) + ATT(I,10)*ATT(I,10)
1885. TSOS = 2.0 * SIGRS
1886. TSOS = 2.0 * SIGDS
1887. SIGR = SIGRS**(.5)
1888. SIGD = SIGDS**(.5)
1889. F = 1. / (SIGR*SIGD*6.2832)
1890. XX = X - XT
```

```
1891.      IF (NFLAG .EQ. 1) GO TO 40
1892.      YY = YT - Y + CT*XX
1893.      D = -YY*G
1894.      R = YY*C - XX/S
1895.      GO TO 50
1896. 40      D = XX
1897.      Z = (YT - Y)*C
1898. 50      AID1 = D*D/TSDS
1899.      IF (AID1 .GT. 10.) GO TO 90
1900.      DD = SFXP(-AID1)
1901.      NCYCLE = NCYCLE + 1
1902.      DO 80  N = 1, NP
1903.      AID2 = R*R/TSRS
1904.      IF (AID2 .GT. 10.) GO TO 70
1905.      DR = SFXP(-AID2)
1906.      DEN = DEN + DR*DD
1907.      NCYCLE = NCYCLE + 1
1908.      IF (KT*ST .GT. 2) WRITE (1005) XT, YT, D, R, DD, DR, DEN
1909. 70      F = F - DLS
1910. 80      CONTINUE
1911. 90      CONTINUE
1912.      DEN = 10000*NS*F*REL*ATT(1,11)*DEN/NP
1913.      DATA(NX) = DATA(NX) + DEN
1914. 100     CONTINUE
1915.      IF (NFLAG2 .EQ. 1) GO TO 110
1916.      NFLAG2 = 1
1917.      WRITE (6, 1001)
1918. 110     CONTINUE
1919.      IF (NX .LT. 17) GO TO 120
1920.      NYT = YT
1921.      IF (NY .LT. 17) GO TO 115
1922.      NX2 = NX1 + 16*INC
1923.      WRITE (6, 1002) (ALOC, NLOC=NX1, NX2, INC)
1924. 115     WRITE (6, 1003) NYT, (DATA(I), I=1, 17)
1925. 120     CONTINUE
1926. 140     CONTINUE
1927.      WRITE (6, 1004) NCYCLE
1928.      RETURN
1929. 160     CONTINUE
1930.      WRITE(5, 1006)
1931.      STOP
1932. 1001     FORMAT('1', 20X, 'EXPECTED HIT DENSITY PER 10000 SQ FT', '/')
1933. 1002     FORMAT(' ', 3X, ' X - LOCATION', '/', ' ', 10X, 17I7, '/',
1934.  X      ' Y LOC  ')
1935. 1003     FORMAT(' ', '/', ' ', 16, 4X, 17F7.3 )
1936. 1004     FORMAT(' ', 3X, 20X, 'CYCLES IN GRIDEN ', I8)
1937. 1006     FORMAT(' ', 3X, ' COMPUTATION STOPPED - CBU WEAPONS ARE',
1938.  X      ' NOT PERMITTED WITH SUBROUTINE GRIDEN')
1939. 1005     FORMAT(' ', 4F6.0, 3E12.4)
1940.      END
```